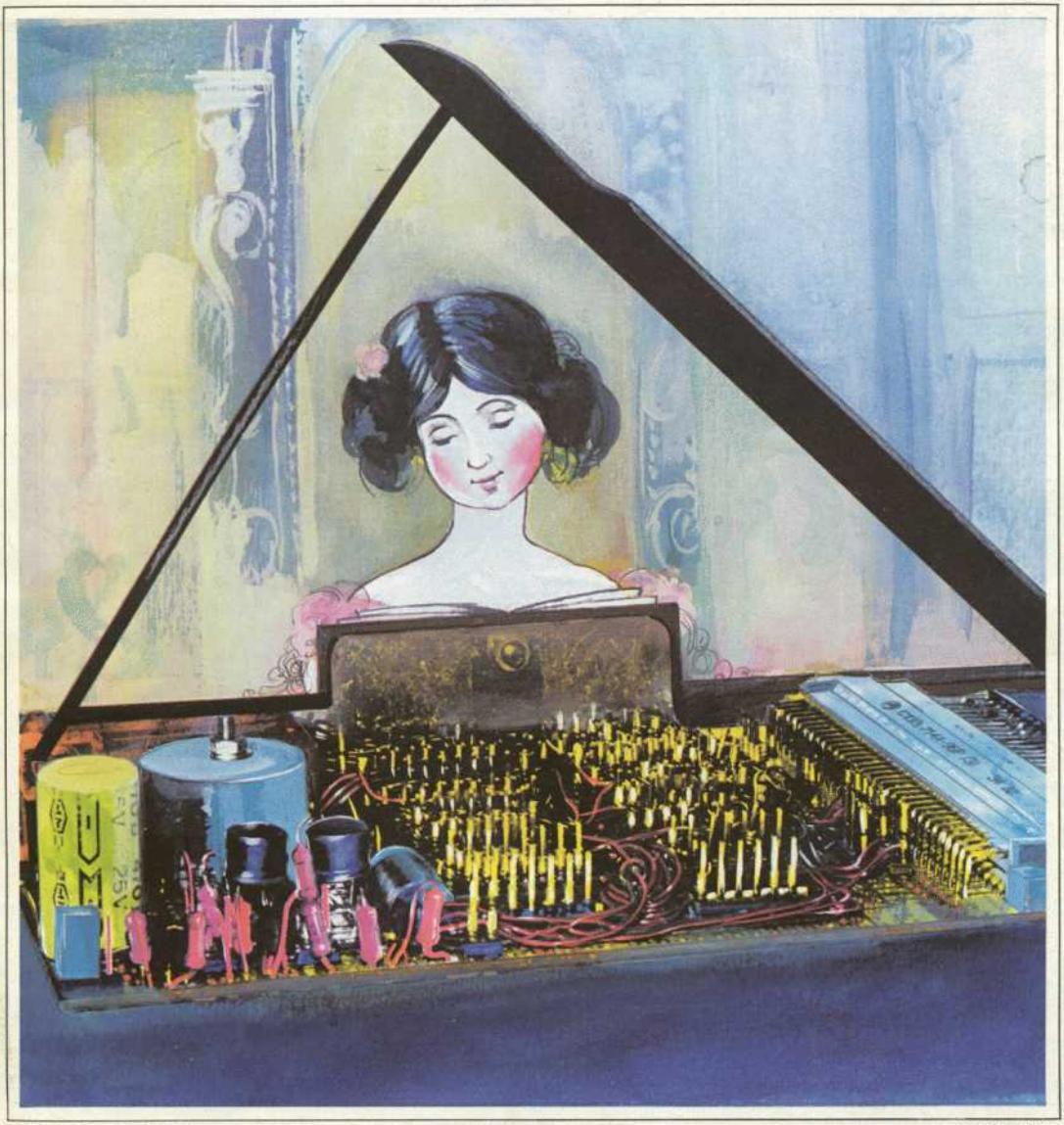
μ mikro számítógép magazin

Ára: 30 Ft



OPUS 27XX

1987/11

Egyedi és általános szoftverek a vállalati ügyvitel területén. Termelésirányítási, raktárgazdálkodási programcsomagok egyedi és hálózatos gépeken.

A szoftvereket kívánság szerint, megrendelőink igényeinek és a helyi viszonyoknak megfelelően átalakítjuk, adaptáljuk. Helyzetfeltárással, rendszertervezéssel, betanítással, a megfelelő számítógépes konfiguráció kiválasztásával és beszerzésével segítünk vállalati problémáik megoldásában.

A garanciális idő letelte után átalánydíjas szervizszolgáltatást vállalunk számítógépeikre.

Egyedi fejlesztések, tervezések és témaorientált feladatok megoldásával is forduljanak hozzánk.

MIKROSZERVÍZ Számítástechnikai Műszaki Fejlesztő

1141 Budapest, Kőszeg u. 4. Kisszövetkezet

Telefon: 831-805

Bankszámla: MHB 207-57 821

5. ÉVFOLYAM
1987/11. SZÁM

μ mikro számítógép magazin

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztőbizottság vezetője:
Kovács Győző

E számunkat szerkesztették:

Bakos Tamás
(programozástechnika)

Broczkó Péter
(hírek)

Kovács Győző
(levelezés)

Lindner László
(sakkprogramozás)

Petróczy Judit
(könyvek)

Simonyi Endre
(klub)

Varga András
(iskola-számítógép)

Címképünk:
Ramocsai Imri munkája

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Petrus György
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlapelőfizetési és Lapellátási Irodáján
(1900 Budapest V., József nádor tér 1.)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, pf. 149.
és a Magyar Média
1932 Budapest, pf. 279.
86—0253

Szikra Lapnyomda
Budapest (87-1520)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Tartalom

Műszaki emlékeink 2
OKTA-TOTÓ 11
Miért nem stupid a MUPID? 24
Nyomtatók holnapja 26
Rendszerfejlesztési eszközök. Szoftver 28
Ígéretes kezdet 29
Még egyszer az Enterprise-ról 30
μINFORM 37
Az Olivetti nem hibázik 39
Adok — veszek — cserélek 39
Olvastunk . . . 40
Petike kártyázik 43
A természet vigyáz ránk 44

ISKOLA-SZÁMÍTÓGÉP

MINI PLUS-Comp BASIC fordító 3
A ROM-ban tárolt aritmetikai rutinok használata 3
TechnoMIR 5

DIÁKROVAT

Betöltő, darazsak, energiaeloszlás 6

PROGRAMOZÁSTECHNIKA

BASIC és gépi kód 12
Z80 programok haladóknak Spectrumra és Primóra 13
Verem a Spectrum BASIC-hez 14
A Spectrum másolói, másolók spektruma 15
Számítógépes grafika Pascalban. Egyenes 17
Hangos billentyűzet 20
Lebegőpontos tömbök átadása. BASIC-Pascal 21

μPROGRAMOK

Memóriaolvasó program 31
Felhasználói karakterek 31

μKLUB

2—4 kbájtos EPROM-égető a Spectrumhoz 32
Integrált szoftver 36

SAKKPROGRAMOZÁS

A játékfa és kiértékelése 42

AZ OLVASÓ ÍRJA 45

KÖNYVEK 47

HÍREK, ÉRDEKESSÉGEK 48

Műszaki emlékeink

„Világos, hogy az olyan problémák, amelyeknek tipikus példája a numerikus időjárás-előrejelzés, teljességgel megoldhatatlanok volnának elektronikus számítógépek nélkül. Éppen ez a modern gépek lényege. Nemcsak egyszerűen meggyorsítják az ember vagy az elektromechanikus gép által elvégzett unalmas, fárasztó és hosszadalmas számításokat. Olyasmit tesznek lehetővé, amit korábban egyáltalán nem lehetett volna megcsinálni. Az elektronikus elv sokkal többet tett, mint csak megszabaditotta az embert attól, hogy rabszolga módjára órákat vesztegessen el a számítások elvégzésére; általa olyasmi is lehetővé és végrehajthatóvá vált, amit az ember egyedül sohasem lett volna képes elvégezni. Ez tette lehetővé, hogy emberek lépjenek a Holdra — és ennek segítségével sikerült másokat épségben visszahozni a Holdra irányuló sikertelen útról."

(H. H. Goldstine: A számítógép Pascaltól Neumannig)

Úgy emlékszem, hogy egyik írásom élén egyszer már idéztem H. H. Goldstine gondolatait. Most azért teszem újra, mert — ha erre egyáltalán szükség van — bizonyítani szeretném, hogy a számítógép nemcsak egy a világ számos műszaki alkotása közül, hanem éppen az egyik legfontosabb: a számítógép révén indult el a XX. század második felének az a szédületes műszaki, intellektuális, gazdasági, sőt társadalmi fejlődése, amelyet méltán neveznek a mai kor ipari forradalmának.

Akik ismernek, tudják, hogy minden iránt, ami számítógép, számítástechnika, informatika, mérhetetlenül elfogult vagyok. Tehát ebben a rövid írásban se várjanak tőlem mást, mint szubjektivitást.

Számítástechnikai műszaki emlékeinket vagy 15 éve próbáljuk megmenteni a következő generáció számára. Akkor határoztuk el ugyanis: nem hagyjuk, hogy a kiöregedett számítógépek a MÉH-telepre jussanak, és így örökre eltűnjenek a semmiben, hanem elraktározzuk valahol, és ha lesz egyszer egy olyan országos műszaki múzeum, amely a műszaki emlékeket ki is tudja állítani, akkor ezeket a gépeket bemutatjuk a nagyközönségnek.

A gyűjtőmunka megkezdődött. Erről dr. Muszka Dániel részletesen beszámolt a μM 1986. márciusi számában. Ő vállalta a Nagyalföldi Kőolaj- és Földgáztermelő Vállalat algyői telepének raktárában tárolt számítógépek szakszerű gondozását és nyilvántartását is. Cikkéből kiderült, hogy egyre nehezebb a gyűjtők munkája, hiszen a korlátozott területű raktárban a hatvanas évek ma már kiöregedettnek számító ICT, Elliott, Siemens, Minszk, Razdan és egyéb gépei mellett egyre kevesebb a hely a folyamatosan selejtezett újabb berendezések számára.

Azt szerettük volna, hogy a szakma élvonalába tartozó vállalatok, kutatóintézetek, egyetemek, főiskolák vállalják el egy-egy öreg számítógép egyes darabjainak megőrzését: készíttessenek egy-két üveges szekrényt, és azokban állítsák ki a gépeket márvánnyal burkolt, elegáns belépőikben, előcsarnokaikban. Ezáltal nemcsak a tárolóhelyet tudnánk növelni, hanem sok ember számára lehetővé tennénk, hogy megismerje a közelmúlt egykor nagy teljesítményűnek tartott számítógépeit is. *Erre a kérésre azonban akkor egyetlen hazai intézmény sem reagált.*

Most, ahogy mondani szokták, van egy jó és egy rossz hírem.

A *jó hír*, hogy néhány lelkes szakember támogatásával talán még az idén létrejön két állandó számítástechnika-történeti kiállítás: az egyik a budapesti Landler Jenő Gimnáziumban, a másik a hódmezővásárhelyi Kossuth Szakközépiskolában. Ezekben a „számítógépes középkádereket" képző intézményekben felismerték annak fontosságát, hogy az odajáró tizenéves gyerekek ne csak BASIC programot tudjanak írni személyi számítógépre, hanem ott az iskolában tanulmányozhassák a számítógépek és számolóeszközök fejlődését is.

Az iskolák vezetőinek egyetértésével úgy tervezzük, hogy a kiállítások nyilvánosak lesznek, tehát amikor nincs tanítás, például a hétvégeken, bárki látogathassa azokat. A diákok vállalják a gépek felügyeletét, és szakszerű magyarázattal szolgálnak az érdeklődőknek. Ha nem tévedek, ez a két iskola elsőként vállalkozik ilyen fontos kulturális szolgálatra, egy állandó technikatörténeti bemutató üzemeltetésére.

A *rossz hírem* az, hogy még ezzel sem enyhülnek lényegesen a raktározás gondjai. Az Országos Műszaki Múzeum vezetőségével újabb számítógépek múzeumi átvételéről folytatott tárgyalásunk sajnos eredménytelen volt. A szűkös elhelyezési lehetőségekre hivatkozva az az álláspontjuk, hogy nagyon megválogatják, milyen darabokat őrizzenek meg a kiöregedett számítógépekből; arról szó sem lehet, hogy teljes konfigurációk raktározását megoldják.

Ezzel a véleménnyel azonban nem lehet egyetérteni. Már csak azért sem, mert nézetem szerint föltétlenül elhibázott szelekciós szempont a helyhiány, hiszen egy régi darab technikatörténeti értékét nem az általa a raktárban elfoglalt terület mérete határozza meg, hanem valami egészen más. Ha arra gondolok, hogy nincs a gyűjteményben EMG 803-as, hogy a KFKI gépeiből csak egyetlen TPA—8-ast, a Videotontól csak az R10-est őrzik, és nincs a régi lyukkártyás rendszereinkből sem . . . Pedig el kellene raknunk a SZTAKI, az SZKI, a SZÁMALK, a KFKI esetleg be sem fejezett kísérleti eszközeit is. Szóval kellene a hely és kellenek a gépek is.

Így megint a Neumann Társaságra marad az újabb gépek gyűjtése és tárolása, de ezt a feladatot csak a műszaki értékekért felelősséget érző és vállaló intézményekkel együtt tudja megoldani.

Ismételt felhívásunkat *kultúrházaknak, művelődési központoknak, oktatási intézményeknek, főleg egyetemeknek és főiskoláknak, nem kevésbé a hazai számítástechnikai gyártás és kutatás fellegvárainak, kutató- és fejlesztőintézeteknek* címezzük. Azt kérjük, vállalják el a gyűjtemény egyes darabjainak kiállítását és megőrzését.

Kérjük továbbá mindazokat, akiknek régi gép van a birtokukban, és azt ki akarják selejtezni, értesítsék a NJSZT titkárságát, hogy ezeknek a becses műszaki emlékeknek a megőrzéséről együttesen tudjunk gondoskodni. És mivel a gépek szállítása is problémát jelent, azoknak az intézményeknek a jelentkezését is várjuk, amelyek ezt vállalnák.

De nemcsak a gépeket szeretnénk megőrizni, hanem egyéb informatikai emlékeket is: programleírásokat, érdekesebb programfuttatások eredményeit, grafikákat, fényképeket, adathordozókat is. Mindezekből igen kevés maradt meg, de talán valaki megtartott néhányat emlékül. Kérjük ezeket is, annak reményében, hogy egyszer lesz majd egy informatikai múzeumunk is, ha nem is olyan szép és nagy, mint a londoni, a müncheni, a bostoni vagy éppen a bécsi, de mindenképpen a magyar műszaki múlt, a hazai informatikatörténet egy darabjának kiállítóhelye.

Be kell vallanom, egy kicsit irigykedem, hogy ezek a gyönyörű műszaki gyűjtemények éppen az iparilag legfejlettebb országokban vannak. Valaki azt mondta: azért, mert ezeknek az országoknak sok pénzük van, így arra is jut, hogy egy halom „ócskavasnak" külön múzeumot építsenek. Bennem viszont az a kérdés ötlött fel: vajon nem azért olyan gazdagok és fejlettek-e ezek az országok, mert még arra is képesek voltak, hogy műszaki alkotásaikat — azt a bizonyos halomnyi „ócskavasat" — a jövő generációjának, tanulságul, megőrizzék.

KOVÁCS GYŐZŐ

MINI PLUS-Comp BASIC fordító

Talán mire megjelennek ezek a sorok, már nem is számít újdonságnak a hír: elkészült a PLUS-Comp, a Plus/4-es BASIC fordítóprogram kazettás verziója. (A lemezes változatot lapunk ez évi 4. számában ismertettük.)

Tudomásom szerint ez az első olyan Commodore BASIC fordítóprogram, amely nem mágneslemezzel, hanem kazettával működik. Bízom benne, hogy nem az utolsó, de erről majd később . . .

Miben különbözik a MINI a PLUS—Comp-tól? Elsősorban a lefordítható BASIC program nagyságában:

	Lemezes	Kazettás
A BASIC sorok max. száma	1600	310
A vezérlésátadó utasítások száma	2000	400
A program-méret	kb. 25 kbájt kb. 12 kbájt	15 kbájt (nem graf.) 7 kbájt (grafika)

A másik különbség a munkaterület helye. A PLUS—Comp ugyanis mágneslemezen tárolja a munkafájlokat, a MINI esetében pedig a memóriában készül a futtatható P-kódú program. Kazettára csak a hibafájl íródik, ha ezt a fordítás előtt kérte.

A fordítóprogram, hasonlóan a „nagyobb testvérhez", elsősorban a CBM 2.0-ás BASIC verziót támogatja, de direktívák segítségével a 3.5-ös verzió is beépíthető a P-kódú programba.

Mivel a fordító felülről lefelé (top-down), balról jobbra (left-right), visszalépés nélküli algoritmussal dolgozik, ha az elemzett BASIC kifejezés legelső tagja nem BASIC 3.5-ös utasítás, direktívával meg kell jelölni, hogy a fordító helyesen építse be a P-kódú programba. Az így megjelölt sor nem tartalmazhat vezérlésátadást és FOR . . . NEXT ciklust (FOR, NEXT, STEP).

A direktívában álló kifejezés vagy sor szintaktikus elemzése csak futás közben folyik. Így fordításkor ezekben a részekben még maradhat hiba. Általánosan igaz, hogy csak alaposan tesztelt programokat érdemes lefordítani.

A lefordítandó BASIC programban a fordító nem használhatja az alábbi utasításokat:

CONT, LIST = a P-kódból származó korlátozás
ELSE
DO UNTIL/WHILE EXIT LOOP
RESTORE n
TRAP/TRON/TROFF/RESUME

A fejlett programozási technikáról való lemondás az ára a mintegy 4—12-szeres sebességnövekedésnek. A legnagyobb sebességet egész típusú FOR ciklusba ágyazott GOSUB-oknál mértem: ez kb. 25-szörös volt a BASIC-hez képest.

A fordító ezek alapján Commodore 64-es programok Plus/4-re átírt változatainak fordítására kiválóan alkalmas. A direktívakezelés elsajátítása kezdő programozónak több időbe telhet, mint azoknak, akik már dolgoztak fordítóprogramokkal. Ezt a kellemetlenséget egy készülő előfordító fogja megszüntetni, amely a lefordítandó BASIC programot automatikusan fordítható állapotba alakítja.

Tervezés alatt áll egy 15 kbájtnál nagyobb programok fordítására alkalmas Plus/4-es fordító, amely a munkafájlokat kazettára írva hozná létre a P-kódú programot. Természetesen ez több időbe telik, mint a memóriában való fordítás. Ha igény lesz rá, TURBO MPC néven kerül majd forgalomba.

Egy bonyolult, sok feltételt tartalmazó program futási ideje BASIC-ben 5 óra volt, ami a lefordított program esetében majd 30 percre csökkent. Ekkora különbséget már mindenképpen meg lehet érezni!

Mivel a MINI fordító készítésénél sok olyan problémát meg kellett oldani, amely fellépne Commodore 64 és C-1530 esetében is, ha megfelelő igény lesz rá, a C64-es verzió elkészítése is szóba jöhet. A leendő fordító érdekessége, hogy a lefordított program a BASIC munkaterületből kevesebb helyet foglal, mint BASIC-ben, függetlenül a BASIC program nagyságától — ellentétben a közismert C64-es BASIC fordítókkal, ahol a szükséges RUN—TIME modul egysoros program esetén is a program elé generálódik, minimálisan 17 blokknyi helyet foglalva a BASIC munkaterületből.

BÁRÓ CSABA

A ROM-ban tárolt aritmetikai rutinok használata

TVC

A TV—Computeren az assembler hiánya miatt csak rövid gépi kódú programok írására vállalkozhatunk. A sokat használt ciklusmagok gépi kódban való megírásával azonban csökkenthető a futási idő, és olyan programok is futtathatók lesznek, melyek tisztán BASIC-ben megírva igen lassúak. A ROM adatmozgató és aritmetikai rutinjaival gépi kódú programunkat szubrutinhívások sorozatává rövidíthetjük. Ezen a módon tetszőlegesen bővíthető a beépített függvénykészlet is.

Amint a gép leírásából kiderül, a TV—Computer az aritmetikai műveleteket a BASIC veremben végzi, és ez a helye a beépített függvények közötti paraméterátadásnak is. A verem utolsó elemére az IY regiszter mutat. A programírás során — különösen ciklusoknál — mindig figyelni kell ennek helyes állítására.

A lebegőpontos számokat a gép háromféle számábrázolási formátum szerint kezeli, amit az ábrákon szemléltetünk. Az ábrázolás bájtonként történik. Egy mantissza-bájt két BCD számjegyet tartalmaz.

Az *1. ábrán* a verem látható. A számok magukban foglalják a túlcsordulásbájtot, amellyel együtt a verembe elhelyezett számok 14 decimális számjegy pontosságúak (egy bájt = két decimális számjegy).

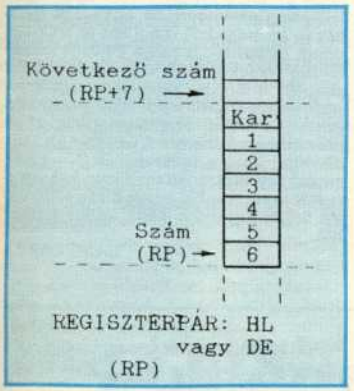
Az ún. regiszter formátumú számoknál nincs túlcsordulásbájt *(2. ábra)*. A gép két beépített aritmetikai regiszterét (X és Y) a beépített függvények használják; a processzor HL és DE regiszterpárjainak megfelelő állításával több ilyen regiszter kialakítható a gép memóriájában.

A *3. ábra* az X és Y belső aritmetikai regiszterek felépítését, a *4. ábra* a szimbólumtáblázatban (változótáblázatban) levő számok formátumát mutatja be. Ez utóbbiak már csak 10 decimális számjegy pontosságúak, ami egyben a megjelenítés pontossága is. A különböző formátumú számok közötti konverziót a ROM adatmozgató rutinjai végzik, a szubrutinhívások során erre nem kell ügyelni.

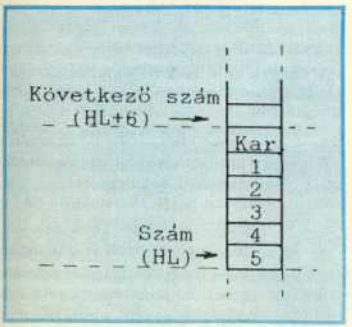
A szubrutinok vagy a szokásos módon, CALL utasítással, vagy táblázatból való kijelöléssel hívhatók. Ilyenkor a szubrutin táblázatbeli sorszámát kell megadni az alábbiak szerint:

RST 18H	A táblázat aktivizálása
sorszám 1	Az 1-es sorszámú szubrutin végrehajtása
sorszám 2	A 2-es sorszámú szubrutin végrehajtása

1. ábra. Az aritmetikai verem



2. ábra. Egy tetszőlegesen kialakítható aritmetikai regiszter



3. ábra. Szimbólumtáblázat-elem

.
.
.

sorszám n Az n-edik sorszámú szubrutin végrehajtása után a szubrutinlista végét az jelzi, hogy a sorszám legnagyobb helyiértékű (128-as súlyú) bitje 1-est tartalmaz. A processzor az ezután következő kódot már utasításként értelmezi, és végrehajtja.

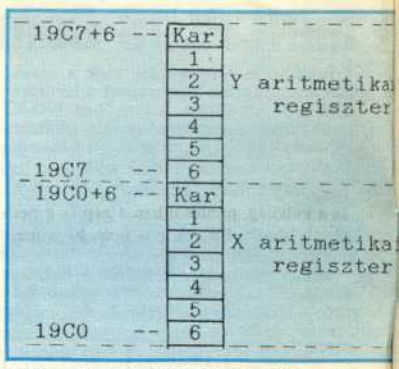
A hívható adatmozgató szubrutinokat a *táblázat* tartalmazza.

A szubrutin hívása	Hová	Honnan	Megjegyzés
CALL EAA2H	Verem (IY)	Regiszter (HL)	IY=IY−9
CALL EAC6G	Regiszter (DE)	Verem (IY)	IY=IY+9
CALL EAD5H	Regiszter (DE)	Verem (IY)	IY marad
CALL FA63H	Verem (IY)	Szimbólumtábla (HL)	IY=IY−9
CALL FB28H	Szimbólumtábla (HL)	Verem (IY)	IY=IY+9
RST 18H 06H	Verem (IY)	X regiszter	IY=IY−9
RST 18H 07H	Verem (IY)	Y regiszter	IY=IY−9
RST 18H 08H	X regiszter	Verem (IY)	IY marad
RST 18H 09H	Y regiszter	Verem (IY)	IY marad
RST 18H 0AH	X regiszter	Verem (IY)	IY=IY+9
RST 18H 0BH	Y regiszter	Verem (IY)	IY=IY+9

Példaként az alábbi programrészlet megcseréli az X és Y aritmetikai regiszterek tartalmát:

```
RST
18H
06H   X regiszter tartalma a verembe
      (IY=IY−9)
07H   Y regiszter tartalma a verembe
      (IY=IY−18)
0AH   Veremből az X regiszterbe
      (IY=IY−9)
8BH   Veremből az Y regiszterbe
      (IY=eredeti érték)
```

A 8BH hatásában megegyezik 0BH-val, de a szubrutinhívások végét jelzi.



4. ábra. A beépített X és Y aritmetikai regiszterek

A táblázatban nem szereplő további adatmozgató rutin, az

```
RST 18H
0CH
```

hatására a veremben levő szám még egyszer rákerül a veremre, önmaga fölé, az IY pedig a másolt számra mutat.

Végül az utolsó adatmozgató rutin a ROM-ban tárolt, megadott sorszámú konstanst a verembe teszi. A sorszámok 0—39-ig terjedhetnek; a hozzájuk tartozó konstansok értéke az *1. listán* levő programmal határozható meg.

A következő programrészlet a veremben egymás fölé helyezi a 4-es, 23-as és a 16-os decimális sorszámú konstansokat:

```
RST 18H
  05H   A szubrutin sorszáma
  04H   A 4-es sorszámú konstans a verembe kerül
        (IY=IY−9)
  05H   A szubrutin sorszáma
  17H   A 23-as sorszámú konstans a verembe kerül
        (IY=IY−18)
  85H   A szubrutin sorszáma befejezéskor
  10H   A 16-os sorszámú konstans a verembe kerül
        (IY=IY−27)
  ...   A processzor ezt a kódot már utasításként kezeli
```

A *2. lista* az USR függvény felépítését mutatja.

BOTTYÁN ZOLTÁN

1. lista

```
 10 LOMEM 8000 : C=0 : D=0
 20 FOR A=0 TO 7
 30 READ B:POKE 7000+A,B:NEXT
 40 C=VARPTR(B)
 50 PRINT"Sorszám,Konstans"
 60 FOR A=0 TO 39
 70 POKE 7003,A: D=USR(7000,C)
 80 PRINT A,B
 90 IF INKEY$="" THEN 90
100 NEXT
110 DATA 229,223,133,39,225,
        195, 40,251
```

2. lista

```
          1B58H   PUSH HL
                  RST 18H
                  85H
POKE 7003,A→      ...
                  POP HL
                  JP FB28H
```

TechnoMIR

Egy interfészrendszernek az a feladata, hogy kapcsolatot teremtsen a számítógép és a külvilág, pontosabban a gép és a perifériák között. Ez akkor is igaz, ha a megszokott perifériákra (képernyő, nyomtató stb.) gondolunk. Így aztán, ha van egy számítógépünk, biztosan kapunk hozzá legalább egy interfészt: azt, amelyik a gépet és a billentyűzetet kapcsolja össze.

Miért mondtuk tehát az előző részben azt, hogy a felhasználás minősége nem a számítógép típusának függvénye? Azért, mert így van! Ezek az interfészek több értelemben is specifikáltak: számítógép-specifikusak (erre még visszatérünk) és periféria-specifikusak.

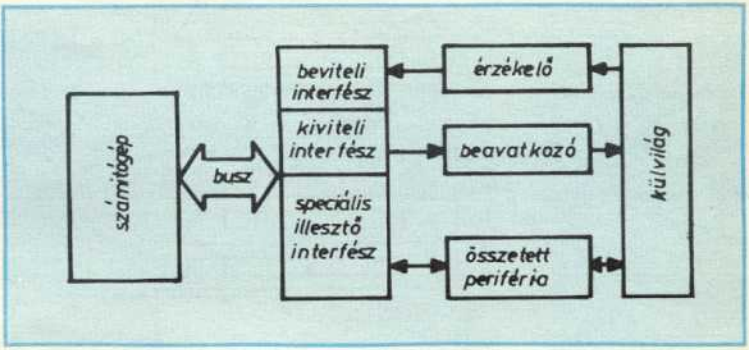
Ez utóbbi érthető. Minél „emberszabásúbb" egy periféria, fizikailag annál összetettebb a feladata. Például a billentyűzet nem egy, hanem kb. 60 kapcsoló; működtetését bonyolult elektronika, az erre szolgáló interfész vezérli. Ez nyilvánvalóan nem lehet azonos a tévékészüléket vagy monitort működtető áramkörrel.

Ennek alapján kimondhatjuk: mivel egy interfész egy és csak egy összetett perifériát tud kezelni, minél egyszerűbb a periféria, annál többféle interfész képes azt kiszolgálni, és megfordítva: egy interfész annál többféle perifériát kezelhet, minél egyszerűbbek azok. (Ez ellen hatnak a szabványosítások, de nagy számuk miatt az állítás — sajnos — jó megközelítése a valóságnak.)

Így aztán, ha számítógépünkkel közvetlenül a külvilág információit akarjuk feldolgoztatni, szükségünk van egy vagy több érzékelőre és az ezekhez alkalmazható interfészre. Ez elég ijesztően hangzik, de minél egyszerűbb az érzékelőnk, annál egyszerűbb a rendszer. Az egyszerű érzékelő jeleit viszonylag egyszerű felépítésű, és az előbb kimondott állítás szerint — ez itt a lényeg! — több érzékelőhöz használható interfész képes feldolgozni. Előny az egyszerűséghez általában tartozó olcsóság, hátrány a pontatlanság. Ez az a kategória, amire jogosan mondják: iskolai demonstrációs célokra kitűnő.

Ide tartozik a TechnoMIR rendszer is, különösen ha a másik előnyét is szem előtt tartjuk: az iskolába telepített géptípusok közül a legtöbbhöz csatlakoztatható. A rendszer elvi és fizikai felépítését az *1.* és a *2. ábra* szemlélteti.

Jól látható, hogy a TechnoMIR rendszer a számítógépeket két csoportba sorolja: az egyikhez szükség van egy speciális modulra is. A hozzáértőbbek látják, hogy a processzor fajtája határozza meg a csatlakozás módját. A rendszer Z80 alapú gépekhez, konkrétan a HT-gépekhez készült. Az, hogy a másik elterjedt processzorcsaládhoz is illeszthető, szintén az egyszerűség következménye: főleg az igen egyszerű és szellemes címzési módé.



1. ábra

Mivel a különböző számítógépek különböző memóriacímeken engedélyeznek felhasználói bővítéseket, az egységek címeit a HT-gépeken használt címekkel adjuk meg. A többi számítógépen használandó címzés:

Primo: HT+192
ZX-Spectrum: 32*HT+31
Commodore 16: HT+65024
Commodore Plus/4: HT+65024
Commodore 64: HT+56832

Például a következő részben részletesen tárgyalt DINP egység HT-címe 24. Primón ez a cím 216, mert 24+192=216.

Jelenleg a készletben a következő egységek kaphatók:

● Külső óra, HCLK (hardware clock). Csak Z80 alapú gépekhez illeszkedő, programozható számláló. Külső óraként is használható.

● Tápegység, POW (power). Ha a számítógép belső tápegysége a kívánt feladat ellátására alkalmas összeállítást nem bírná el, külső táplálást biztosít.

● Digitális bemenet, DINP (digital input). 8 darab TTL-szintű bináris jel fogadására alkalmas, aktív bemenetekkel rendelkező egység.

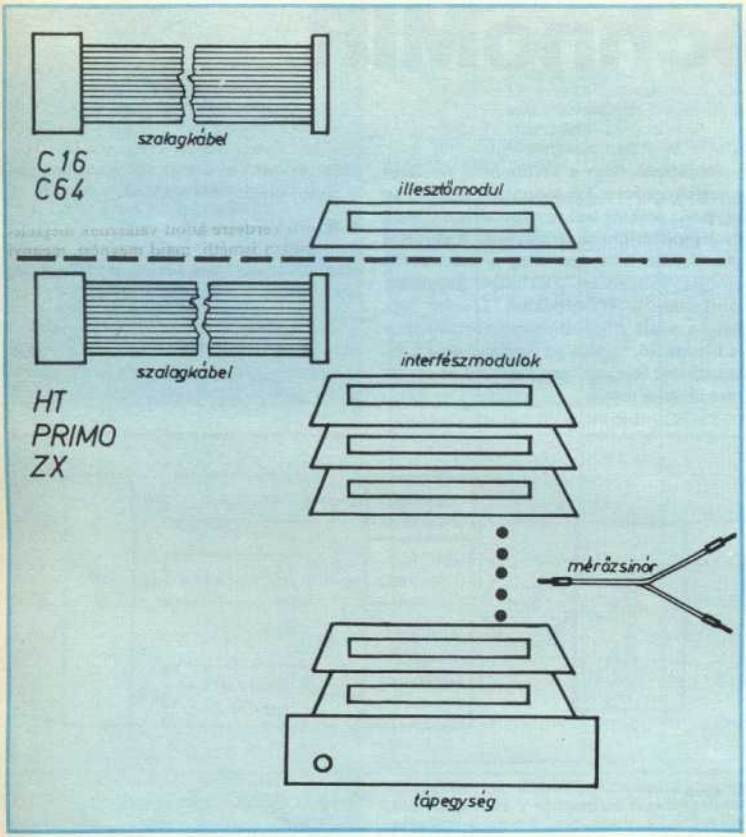
● Digitális kimenet, DOUT (digital output). 8 darab TTL-szintű bináris jelet szolgáltató, több változatban kapható, aktív és passzív kimeneteket tartalmazó egység. Változatai az aktív kimenetek számában különböznek.

● Megszakítást kezdeményező bemenet, ITIN (interrupt input). Ugyanolyan, mint a DINP, de a beérkező jel megszakítást kezdeményez. Gépi szintű programozási ismereteket igénylő egység.

● Digitális—analóg átalakító, D/A (digital/analog converter). 3 darab, egyenként 10 volt feszültségtartományban programozható kimenetet tartamazó egység.

● Analóg—digitális átalakító, A/D (analog/digital converter). Több változatban kapható, meghatározott feszültségtartományon belül digitális feszültségmérésre alkalmas egység. Érdekessége, hogy a mérési tartomány a program segítségével elcsúsztatható.

● Analóg sokszorozó, AMUX (analog multiplexer). Hat csatornán analóg jeleket képes fogni, és programból kiválasztható, hogy melyik jelet továbbítsa az A/D egységnek.



2. ábra

● Próbapanel, PB (proto board). 3 × 8 bit programozható ki- vagy bemenettel és tápfeszültségpontokkal ellátott, speciálisan kialakított felületű egység. Egyszerűbb digitális kapcsolások építhetők vele, amelyek a számítógéppel vizsgálhatók is.

● Botkormányillesztő, JOY (joystick). Olyan géphez való, amelynek nincs ilyen csatlakozója.

Tudomásunk szerint több érdekes modul áll fejlesztés vagy gyártás-előkészítés alatt; például az a léptetőmotor-kezelő modul, amely a készülő iskolarobotot fogja az iskolai számítógépek segítségével vezérelni.

Az általános áttekintés után a következő részben elkezdjük a modulok konkrét ismertetését.

ALBU LÁSZLÓ—KIRÁLY LÁSZLÓ

Minden hétfőn 17-től 19 óráig

a Mikroszámítógép Magazin munkatársai és felkért szakértők válaszolnak az olvasók kérdéseire a szerkesztőségben: Budapest II., Fő u. 68. I. em. 109. vagy a 154-090 és a 154-250-es telefonon.

Minden kedden 17-től 20 óráig ENTERPRISE-klub a VSZM Közösségi Házban

(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor

DIÁK — COMMODORE—64 — ROVAT

Betöltő

Akik szeretnek Simon's BASIC-ben programozni, tudják, hogy elég bonyolult ezzel a nyelvvel dolgozni: minden alkalommal be kell tölteni a Simon's BASIC-et, elindítani, majd betölteni a programot és ezt is elindítani. A betöltő mindezt magától elvégzi:

1—7 A menü képernyőre írása. Szövege természetesen más is lehet
8 A billentyűzetről csak az 1-est vagy a 2-est fogadja el
10 A puffer maximálisra állítása
20 Adatok betöltése a billentyűzetpufferba
30 A képernyőre kikerülő betűk ASCII kódja
40 A Simon's betöltése

A programokat 0-tól 9-ig sorszámozva lehet kimenteni lemezre. Például: SAVE"6",8. Természetesen a menüben ehhez a számhoz is kell egy programnevet rendelni. Úgy is átalakíthatjuk, hogy nem számot, hanem karakter nevű programot töltsön be:

8 IF A4="6" THEN A4="D"

A lényeg az, hogy a program neve csak egy karakter lehet.

Darazsak

A gimnáziumok I. osztályos fizikájához kapcsolódó program a részecskék eloszlását szimulálja 8 darázzsal. Két szobában 4-4 darázs van. A két szoba között kinyitjuk az ajtót, és a darazsak elkezdenek röpködni egyik szobából a másikba. A program megvizsgálja és kijelzi az eloszlások számát, erről grafikont készít, amit ki is lehet nyomtatni. A program működése:

10—140 Sprite-ok kikapcsolása, kiírások, adatok bekérése
150 GOSUB 580
580—990 A sprite-ok definiálása, a szobák kirajzolása, a darazsak elhelyezése
1000 RETURN
160—240 A röptetendő darázs kiválasztása
200 GOSUB 1010, zümmögés
210 GOSUB 470, röptetés: a bal oldali szoba egyik darazsát a jobb oldalba
220 GOSUB 520, a bal szobából röptetés a jobb szobába
250—290 Az egyes állapotok kiírása
300—400 Grafikonrajzolás
410—440 GOSUB 1100, nyomtatás

```
POKE53280,5
POKE53281,7:PRINT"■■■■■■■■■■■■■■■■■■■■*** FIZIKA ***"
PRINT"■■■■■■■■■■■■■■■■■■■(C) 1987 TOROK GYORGY"
PRINT"■■■■■■■■■■■■■■■■■■■■■■■■ 1 ■ DARAZSAK"
PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■ 2 ■ ENERGIA ELOSZLAS"
PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■ ? ■ VALASSZON!"
GETA$:IFA$=""OR(A$<>"1"ANDA$<>"2")THEN7
0 POKE198,10:POKE643,10:PRINT"■■■■■■■■■■■■■■■■■■■■LOADING..."
0 FORI=0TO9:READA:POKE631+I,A:NEXTI
0 DATA 76,207,34,48,34,44,56,13,82,131
5 POKE634,ASC(A$):PRINT"■"
0 A=0:IFA=0THEN LOAD"SIMONS BASIC",8,1
0 END
```

Betöltő

Energiaeloszlás

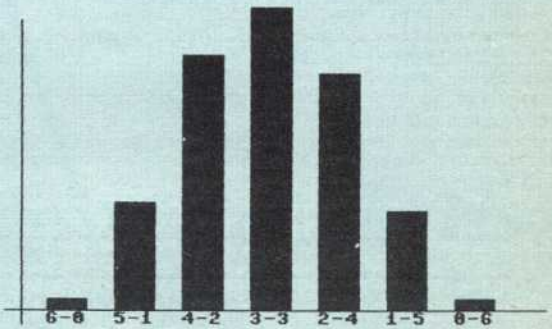
Ez a program is egy fizikai jelenséget vizsgál, a részecskék energiaeloszlását zárt térben. A kísérletben négyzetek jelölik a részecskéket, babok az energiát. Kezdetben minden részecskének egy energiája van. A számítógép kiválaszt egy részecskét, ezt képzeletben nekiütközteti egy másiknak, és az energia (a bab) átrepül. Ezt a 630-as sorban levő kérdésre adott válasznak megfelelő számban ismétli, majd megnézi, mennyi energiájuk van az egyes részecskéknek, és ezt megjegyzi. A kísérlet végén az átlagértékeket táblázatban kiírja, ami természetesen ki is nyomtatható.

TÖRÖK GYÖRGY

Darazsak

```
20 FORI=1TO6:MOB OFF I:NEXTI
40 POKE53281,5:PRINT"■"
50 PRINT"■■■":CENTRE"MONORI JOZSEF ATTILA"
60 PRINT"■":CENTRE"GIMNAZIUM"
70 PRINT"■■■":CENTRE"*** TOROK GYORGY ***"
80 PRINT"■■":CENTRE"(C) 1986"
90 PRINT"■■":CENTRE"*** DARAZSAK ***"
100 GETA$:IFA$=""THEN100
110 INPUT"■■■■■HANYSZOR ROPTESSEK ■■■■■■■■":V$:V=VAL(V$):IFV=0THEN110
120 INPUT"■■■■■■■MILYEN SEBESSEGGEL (0-255) ■■■■■■■■":S$:S=VAL(S$)
130 IFS<0ORS>255THEN120
140 S=255-S
150 GOSUB580
160 FORI=1TO6:X(I)=0:NEXTI
170 A=6:B=0
180 FORI=1TO3:PRINTAT(15,I+6)" ":NEXT
190 FOR Q=V TO 1 STEP -1
200 G=INT(RND(0)*6)+1:F1=10:F2=0:GOSUB1010
210 IF X(G)=0 THEN GOSUB470:GOTO230
220 IF X(G)=1 THEN GOSUB520
230 PRINTAT(15,22)"    ":PRINTAT(14,22)Q
240 NEXT Q
250 POKEN,0:POKEAA,0:POKEHH,0
260 FOR I=1TO6:MOB OFF I:NEXT
270 PRINT"■■■■■■■■■*** ALLAPOTOK GYAKORISAGA ***■"
280 FOR I=0TO6:PRINT"■■■■■■■■■■■■■■■■"6-I"- "I" :"H(I):NEXTI
290 GET A$:IFA$=""THEN290
300 GOSUB310:GOTO390
310 PRINT"■":POKE53281,15:C=0
320 FORI=1TO6:IFH(I)>Z1THENZ1=H(I)
330 NEXT I
340 Z1=189/Z1:HIRES0,1
350 LINE 10,0,10,200,1:LINE 0,190,320,190,1
360 TEXT0,192,"   6-0  5-1  4-2  3-3  2-4  1-5  0-6",1,1,8
370 FOR K=24 TO 33*8 STEP 5*8:FORN=1TO24:LINE K+N,189-(H(C)*Z1),K+N,189,1
380 NEXTN:C=C+1:NEXTK:RETURN
390 GET A$:IFA$=""THEN390
400 NRM
410 PRINT"■■■■■■■■■■■■■■■■":CENTRE"KINYOMTASSAM AZ ADATOKAT (I/N) ?"
420 GET A$:IFA$=""THEN420
430 IF A$="I" THEN1100
```

```
440 IF A$<>"N" THEN420
450 RUN
460 STOP
470 RLOCMOB G,136,106,0,S
480 RLOCMOB G,152+INT(RND(0)*80),74+INT(RND(0)*80),0,S
490 A=A-1:B=B+1:H(A)=H(A)+1:X(G)=1
500 PRINTAT(7,19)A:PRINTAT(21,19)B
510 RETURN
520 RLOCMOB G,136,106,0,S
530 RLOCMOB G,40+INT(RND(0)*80),74+INT(RND(0)*80),0,S
540 A=A+1:B=B-1:H(A)=H(A)+1:X(G)=0
550 PRINTAT(7,19)A:PRINTAT(21,19)B
560 RETURN
570 STOP
580 POKE53281,5:PRINT"■"
590 DESIGN 1,64*128
600 @.....B......
610 @.....B......
620 @....CBC.....
630 @....CBC.....
640 @....BBB.....
650 @.....B......
660 @...DCCCD....
670 @..DDBBBDD...
680 @..DDCCCDD...
690 @.DDDBBBDDD..
700 @.DDDCCCDDD..
710 @.DDDBBBDDD..
720 @DDDDCCCDDDD.
730 @DDDDBBBDDDD.
740 @DDDDCCCDDDD.
750 @DDDDBBBDDDD.
760 @DDDDCCCDDDD.
770 @DDDDBBBDDDD.
780 @DDD.CCC.DDD.
790 @.D...B...D..
800 @............
810 MOB SET 1,128,7,1,1
820 MOB SET 2,128,7,1,1
830 MOB SET 3,128,7,1,1
840 MOB SET 4,128,7,1,1
850 MOB SET 5,128,7,1,1
860 MOB SET 6,128,7,1,1
870 CMOB 0,3
880 PRINTAT(1,1)"■                    ■"
890 PRINTAT(1,16)"■                    ■"
900 FOR1=2TO15:PRINTAT(1,I)"■ ■                ■ ■                ■ ■":NEXT
910 FORI=1TO3
920 MMOB I,56,74+J,56,74+J,0,0
930 MMOB I+3,96,74+J,96,74+J,0,0
940 J=J+32:NEXTI
950 D$(1)="┌──┐"
960 D$(2)="| 0 |"
970 D$(3)="└──┘"
980 FORI=1TO3
990 PRINTAT(6,17+I)D$(I):PRINTAT(20,17+I)D$(I):NEXTI
1000 RETURN
1010 REM ZENE--------------------
1020 SI=54272:FL=SI:FH=SI+1:W=SI+4:AA=SI+5:HH=SI+6:LL=SI+24
1030 POKEW,0:POKEAA,0:POKEHH,0
1040 POKELL,15
1050 POKEAA,16*5+16
1060 POKEHH,0
```



```
1070 POKE FH,F1:POKEFL,F2
1080 POKEW,33
1090 RETURN
1100 OPEN 4,4
1110 PRINT#4,"     ********************"
1120 PRINT#4,"     ***   DARAZSAK    ***"
1130 PRINT#4,"     ********************"
1140 PRINT#4:PRINT#4
1150 PRINT#4,"      ROPTTESEK SZAMA"V
1160 FORI=0TO6:PRINT#4,"      "6-I"-"I":"H(I):NEXTI
1170 PRINT#4:CLOSE4
1180 COPY
1190 RUN
```

Energiaeloszlás

```
0 PRINT"■":POKE53281,11:POKE53280,11
1 PRINT"■■■■■":CENTRE"*** J.A.G. ***":PRINT"■■":CENTRE"ENERGIA ELOSZLAS"
2 PRINT"■■■■■":CENTRE"*** TOROK GYORGY ***"
3 PRINT"■■":CENTRE"(C) 1986"
5 GETA$:IFA$=""THEN5
7 DIM B(500,7):R$=CHR$(8):T$=CHR$(15)
8 GOSUB 600:PRINT"■"
10 A$(1)="┌──────────┐"
20 A$(2)="|          |"
30 A$(3)="└──────────┘"
40 FORI=0TO2
50 PRINTAT(I*13+1,0)A$(1)
60 PRINTAT(I*13+1,12)A$(1)
70 FORJ=0TO8
80 PRINTAT(I*13+1,J+1)A$(2)
90 PRINTAT(I*13+1,J+13)A$(2)
100 NEXTJ
110 PRINTAT(I*13+1,10)A$(3)
120 PRINTAT(I*13+1,22)A$(3)
130 NEXTI
150 DESIGN 0,13*64
160 @........BBBB............
162 @......BBBBBBBB..........
164 @....BBBBBBBBBB..........
166 @...BBBBBBBBBBBB.........
168 @..BBBBBBBBBBBBB.........
170 @..BBBBBBBBBBBBB.........
172 @.BBBBBBBBB..B...........
174 @.BBBBBBBBB.BB............
176 @.BBBBBBBB.BBB............
178 @.BBBBBBBB.BB............
180 @.BBBBBBB.BBB............
182 @.BBBBBBB.BBB............
184 @.BBBBBBBB.BB............
186 @.BBBBBBBB.BBB............
188 @..BBBBBBBB.BB............
190 @..BBBBBBBBB..B...........
192 @...BBBBBBBBBBB...........
194 @...BBBBBBBBBBB...........
196 @....BBBBBBBBBB...........
198 @.....BBBBBBBBB...........
200 @.......BBBBBB............
210 MOB SET0,13,7,0,0
220 MOB SET1,13,7,0,0
225 MOB SET2,13,7,0,0
230 MOB SET3,13,7,0,0
235 MOB SET4,13,7,0,0
240 MOB SET5,13,7,0,0
245 MOB SET6,13,7,0,0
250 FORI=1TO6:F(I,I)=1:READX(I),Y(I):Z(I)=1
252 MMOBI,X(I),Y(I),X(I),Y(I),0,0
253 NEXT
255 REM---------------------------------
256 REM---------------------------------
300 FORU=1TO01
302 FORU1=1TO02
305 R1=INT(RND(0)*6)+1:R2=INT(RND(0)*6)+1
310 IF Z(R1)=0ORR1=R2THEN305
320 FOR I=1TO6
330 IF F(R1,I)=1 THENH=I:GOTO350
340 NEXTI:GOTO300
350 F(R1,H)=0:Z(R1)=Z(R1)-1
360 Z(R2)=Z(R2)+1:F(R2,H)=1
365 YY=INT(RND(0)*40):XX=INT(RND(0)*40)
370 RLOCMOB H,X(R2)+XX,Y(R2)+YY,0,0
380 NEXTU1:PRINT AT(18,24)U;
390 FORI=1TO6
400 E(I)=E(I)+Z(I)
405 B(U,I)=Z(I):NEXTI
```

```
410 NEXTU:FORI=1TO6:MOB OFF I:NEXTI:PRINT"■":GOTO487
425 PRINT"■";
430 CENTRE"ELOSZLAS":PRINT
435 PRINT"┌──────┬──────┬──────┬──────┬──────┬──────┬──────┐"
440 PRINT"|DOBAS| 1. | 2. | 3. | 4. | 5. | 6. |"
445 PRINT"├──────┼──────┼──────┼──────┼──────┼──────┼──────┤"
450 FORI=1TO16
460 PRINT"|      |      |      |      |      |      |      |"
470 NEXT
475 PRINT"└──────┴──────┴──────┴──────┴──────┴──────┴──────┘"
477 PRINT"■■■■■■■■■";
481 PRINTAT(0,18)"├──────┼──────┼──────┼──────┼──────┼──────┼──────┤"
482 PRINTAT(1,19)"ATLAG"
485 PRINT"■■■■■■":RETURN
487 GOSUB425
490 FORII=1TO01
500 PRINTAT(1,G+6)II*02
510 PRINT"■■■■■■■■■"B(II,1)"■■"B(II,2)"■■"B(II,3)"■■"B(II,4)"■■";
520 PRINTB(II,5)"■■"B(II,6)
530 G=G+1:IFG=11THENG=0:GOSUB1100
540 NEXTII
545 FORI=1TO6:E(I)=INT((E(I)/01)*100)/100
546 PRINTAT(I*5+2,19)RIGHT$("    "+STR$(E(I)),4):NEXTI
547 W8=0:FORI=1TO6:W8=W8+E(I):NEXTI:YU=W8/6
548 PRINTAT(6,23)"AZ ATLAGOK ATLAGA:";INT(YU*1000)/1000
550 GET A$:IFA$<>" "THEN550
560 PRINT"■■■■■■■■■■■■■■■■■■■■KINYOMTASSAM AZ ADATOKAT (I/N)?"
570 GETA$:IF A$="I"THEN2000
580 IFA$<>"N"THEN570
590 RUN
600 REM**********************************
610 PRINT"■"
620 INPUT"■■■■■■■■HANYSZOR DOBJAK";01$:01=VAL(01$):IF01=0THEN620
630 PRINT"■■■■■■■■■■HANYASSAVAL JELEZZEM KI":INPUT"■■■AZ EREDMENYT";02$
635 02=VAL(02$):IF02=0THEN630
640 RETURN
997 REM--------------------------------
998 REM--------------------------------
999 REM--------------------------------
1000 DATA56,58,152,58,260,58,56,154,152,154,260,154
1100 PRINTAT(16,22)"SPACE=>TOVABB"
1110 GETA$:IFA$<>" "THEN1110
1120 GOSUB425:RETURN
2000 PRINT"■";
2005 OPEN 4,4
2010 PRINT#4,"                    ELOSZLAS"
2020 PRINT#4,"┌──────┬──────┬──────┬──────┬──────┬──────┬──────┐";R$
2030 PRINT#4,T$;"|DOBAS| 1. | 2. | 3. | 4. | 5. | 6. |";R$
2040 PRINT#4,T$;"├──────┼──────┼──────┼──────┼──────┼──────┼──────┤";R$
3030 FORII=1TO01
3040 PRINT#4,T$;"|"II*02;
3050 PRINT#4,CHR$(16);"06";"| "B(II,1)"| "B(II,2)"| "B(II,3)"| "B(II,4)"| ";
3060 PRINT#4,B(II,5)"| "B(II,6)"| ";R$
3080 NEXTII
3082 PRINT#4,T$;"├──────┼──────┼──────┼──────┼──────┼──────┼──────┤";R$
3084 PRINT#4,T$;CHR$(16);"06";
3090 FORI=1TO6:E(I)=INT((E(I)/01)*100)/100
4000 PRINT#4,RIGHT$("       "+STR$(E(I)),5);:NEXTI
4002 PRINT#4,CHR$(16);"00";"|ATLAG|      |      |      |      |      |      |";R$
4010 PRINT#4,T$;"└──────┴──────┴──────┴──────┴──────┴──────┴──────┘"
4012 PRINT#4:PRINT#4,"AZ ATLAGOK ATLAGA:";INT(YU*1000)/1000:PRINT#4:PRINT#4
4015 CLOSE4
4020 RUN
```

OKTA-TOTÓ

Nézzük az előző számunkban megjelent és egyben utolsó forduló megfejtését. A kérdéseket általános témákból merítettük: operációs rendszer, programozás, adatbázis-kezelés stb. Aki az első kérdésre 1-est tippelt, majdnem eltalálta a helyes választ, mert a számítógép operációs rendszer nélkül az esetek túlnyomó többségében nem üzemeltethető. Azonban létezik olyan üzemmód és készítenek olyan programokat, amikor a gépet alap operációs rendszer nélkül is lehet használni **(X)**.

A számítógép bekapcsolását követően az operációs rendszert tartalmazó szoftvert a hajlékonylemez-meghajtón keresi, és ha ott nem találja, akkor automatikusan a merevlemezről olvassa. Mivel a hajlékonylemezek elég sérülékenyek, ezért általános, hogy a merevlemezegységgel rendelkező gépeknél az operációs rendszert azon helyezik el, illetve rámásolják. Ez azonban nem zárja ki, hogy hajlékonylemezről is betölthető **(2)**.

A jelenleg gyártott digitális számítógépek működési elve a Boole-algebrán alapul. Nem szándékozunk jóslatokba bocsátkozni, de valószínűleg ez még sokáig így lesz **(X)**.

A hexadecimális, azaz tizenhatos számrendszer tizenhat jelből áll. A számítástechnikában használt általános konvenció szerint az arab számjeleket az angol ábécé betűivel egészíti ki. Így a „G" nem fordulhat elő **(X)**.

Az adatbázis szempontjából a mező a rekordnak az a része, amely az adatfeldolgozás szempontjából egységnek kezelhető. A rekord egymással kapcsolatban álló adatok (mezők) egységként kezelt halmaza. A fájl egymással kapcsolatban álló adatok *szervezett* halmaza **(2)**.

Az ellenőrző szám a hibásan bevitt, begépelt adatok kiszűrésére szolgál. Képzése az egyszerű összeadástól a komplikált logikai elven felépített rendszerig terjedhet. Szűkebb értelemben a paritás-ellenőrzés is ebbe a témakörbe tartozik **(1)**.

A Chapin kártya a folyamatábrához hasonló programozási segédeszköz **(2)**.

A félvezetők és integrált áramköri megvalósításaik a modern számítógépek építőelemei. A logikai áramkörök felépítésénél a félvezetőknek azon tulajdonságait hasznosítják, hogy két állapotuk van. Vagy van jel („igen"), vagy nincs jel („nem") **(2)**.

Szekvenciális adatállományt általában mágnesszalagon hoznak létre. Indexelt szekvenciális adatállomány azonban csak mágneslemezen valósítható meg **(1)**.

A disassembler gépi kódból assembler nyelvű programot állít elő **(X)**.

Az algoritmus olyan szabályokból áll, amelyek egy feladat végrehajtásának egymást követő lépéseit határozzák meg. Algoritmizálható feladatokra minden esetben program is készíthető. Bebizonyították, hogy léteznek olyan matematikai, kibernetikai problémák, amelyek nem algoritmizálhatók **(1)**.

A PS/2 (Personal System) az új IBM gépcsalád elnevezése **(2)**.

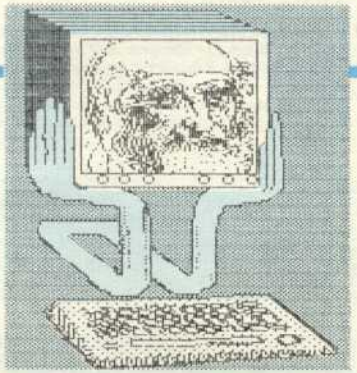
A parancsot bevitelének pillanatában végrehajtja a számítógép (például LIST), az utasítást pedig akkor, amikor a vezérlés rákerül, a program futása közben **(1)**.

Az MS—DOS operációs rendszernél a DOS parancs segítségével vissza lehet térni az alap operációs rendszerbe a futó programból való kilépés nélkül. A programba az EXIT paranccsal térhetünk vissza **(1)**.

A helyesen kitöltött szelvény:

X 2 X X 2 1 2 2 1 X 1 2 1 1

Az ember elgépiesedik —



a gép elemberiesedik

Karikatúra-pályázat

Az ember, az emberi munka elgépiesedésének témaköre a humor egyik kifogyhatatlan ötlettára. Sokan hitték, hiszik ma is, hogy az emberiség — az emberség — egyik nagy ellensége a számítógép. Miközben erről értekeznek, a számítástechnikusok jóvoltából egyre intelligensebb, egyre inkább emberszabású számítógépek, okos programok készülnek. Komoly eredményeket értek el a mesterséges intelligencia kutatásában, s a számítógépek megértik az emberi szót, válaszolnak, kézírást olvasnak, kottából muzsikálnak: partnereink lettek nemcsak a munkában, de a szórakozásban is.

Az ember elgépiesedik — a gép „elemberiesedik". Összecsap a mesterséges intelligencia a természetes értelemmel. Mi lesz ebből a párviadalból? Erre várunk választ a világ karikaturistáitól és rajzos kedvű számítástechnikusaitól.

A pályázatot a Neumann János Számítógéptudományi Társaság hirdeti meg. A kiállítás és díjkiosztás a Tavaszi Fesztivál μ'88 rendezvényei között szerepel, 1988 márciusában.

Két kategóriát hirdetünk meg. Az elsőbe a hagyományos eszközökkel (ceruza, toll, ecset stb.) dolgozó karikaturisták műveit várjuk. A második kategóriába azok a karikaturisták és számítógépes szakemberek, amatőrök jelentkezhetnek, akik műveiket számítógép segítségével készítették.

A pályázók mindkét kategóriában 3-3 alkotásukkal vehetnek részt. A második kategóriában kérjük a szerzőket, hogy műveiket vagy nyomtatva, vagy fényképen (tehát közvetlenül kiállítható formában) küldjék be.

A pályázat nyílt: bárki indulhat, aki művét a határidőig (1987. dec. 31.) elküldi. A képek méretét nem kötjük meg, hiszen annak sokszor a technika szab határokat.

A pályázók nevük, lakcímük mellett írják meg korukat és foglalkozásukat is. A második kategória szerzőitől kérjük a géptípus, illetve a felhasznált program(ok) megnevezését, saját program esetén ennek feltüntetését.

A műveket nemzetközi zsűri értékeli. Egy nagydíjat adunk ki, valamint kategóriánként egy-egy első, második és harmadik díjat. Ezek mellett néhány különdíjat is kiosztunk.

A zsűri által a kiállításra kiválasztott karikatúrákból reprezentatív albumot jelentetünk meg, melyet a benne szereplő pályázók ajándékba megkapnak, mások számára pedig a kiállításon kezdjük meg az árusítását. Az album postán is megrendelhető (SZÁMALK Kiadó, Budapest, Bartók Béla út 104. 1113).

A beküldött karikatúrákat nem küldjük vissza. A kiállítás utolsó napján a műveket árverésre bocsátjuk, s a bevételt testi fogyatékosok számítástechnikai munkalehetőségei megteremtésére fordítjuk.

A pályaműveket az alábbi címre kérjük: Neumann János Számítógéptudományi Társaság, Budapest, Báthori u. 16. 1054.

A kiállítás szervezője: dr. Halász Géza, SZÁMALK. Tel.: 851-011.

BASIC és gépi kód

Legutóbb a BIT és a NOP utasításról volt szó. Most egy újabb utasításcsoporttal ismerkedhetünk meg, a veremkezelő utasításokkal. Ez a csoportosítás egy kissé önkényes, mert általában csak a PHA, PHP, PLA és PLP utasításokat szokták ide számítani, tágabb értelemben viszont a JSR és RTS utasításokat is ide soroljuk.

A veremkezelő utasításokat nagyon körültekintően kell használni, mert a legkisebb gondatlanság is kellemetlen következményekkel járhat, amin csak a gép kikapcsolásával, vagy szerencsés esetben a RESET gomb használatával tudunk segíteni.

A PHA

Neve az angol „PusH Accumulator on stack" kifejezés betűiből származik. Ennek megfelelően az A regiszter (másik nevén akkumulátor) tartalmát a processzorverembe másolja, a veremmutató által meghatározott címre, majd a veremmutató tartalmát eggyel csökkenti. Az állapotbitek értékét nem módosítja.

A PLA

Neve az angol „PulL Accumulator from stack" kifejezés betűiből keletkezett. A veremmutató tartalmát eggyel megnöveli, majd a veremmutató által meghatározott bájt tartalmát az A regiszterbe tölti. A Z és N állapotbitek az A regiszternek az utasítás végrehajtása utáni tartalmától függően állítódnak be az ismert módon.

A PHP

Neve az angol „PusH Processor status on stack" kifejezés betűiből származik. Az állapotregiszter tartalmát a verembe másolja, a veremmutató által meghatározott címre, majd a veremmutató tartalmát eggyel csökkenti. Az állapotbitek értéke nem változik meg.

A PLP

Neve az angol „PulL Processor status from stack" kifejezés betűiből keletkezett. A veremmutató tartalmát eggyel megnöveli, majd a veremmutató által meghatározott bájt tartalmát az állapotregiszterbe tölti.

A TXS

Az X regiszter tartalmát a veremmutató regiszterbe másolja. Az állapotbitek értéke nem változik. Ezzel az utasítással tudjuk a veremmutató értékét módosítani. Alkalmazására az *1. listán* láthatunk példát.

A TSX

A veremmutató tartalmát az X regiszterbe másolja. A Z és N állapotbitek az X regiszter tartalmának megfelelően állítódnak be. Ennek az utasításnak a segítségével tudunk programból a veremmutató tartalmához hozzáférni. Alkalmazására a *2. listán* láthatunk egy példát.

A BRK

Neve az angol „BReaK" (törés, megszakítás) szó betűiből származik. A B és I állapotbiteket 1-re állítja, azután az utasításszámláló és az állapotregiszter tartalmát a verembe teszi, majd az IRQ vektor tartalmát az utasításszámláló regiszterbe tölti. Az IRQ vektor a $FFFE . . . $FFFF címeken található, a megszakításkezelő rutin címét tartalmazza. Erről az I bitet állító utasítások ismertetésekor (Mikroszámítógép Magazin 1987/5. BASIC és gépi kód) már volt szó.

Az RTI

Neve az angol „ReTurn from Interrupt" (visszatérés megszakításból) betűiből származik. A feltételbitek állításától eltekintve ugyanazt csinálja, mint a BRK, de fordított sorrendben és fordított irányban. Az RTS utasítástól csak annyiban különbözik, hogy a veremből először az állapotregiszter korábbi tartalmát tölti vissza, és csak azután tölti az utasításszámlálóba — az RTS-hez hasonlóan — a visszatérési címet.

A programlistákról

A mostani programlistákon a C64 KERNAL ROM-jának egy-egy környezetéből kiemelt részlete látható. Ezek a másik két géptípuson is megtalálhatók, az ittenitől eltérő címeken, a VC20-on pontosan megegyező formában, a C16-on apró — a lényeget nem érintő — eltérésekkel.

```
fce2  a2ff    ldx #$ff
fce4  78      sei
fce5  9a      txs
fce6  d8      cld
```

1. lista

```
ff48  48      pha
ff49  8a      txa
ff4a  48      pha
ff4b  98      tya
ff4c  48      pha
ff4d  ba      tsx
ff4e  bd0401  lda $0104,x
ff51  2910    and #$10
ff53  f003    beq $ff58
ff55  6c1603  jmp ($0316)
ff58  6c1403  jmp ($0314)
```

2. lista

3. lista

```
ea81  68      pla
ea82  a8      tay
ea83  68      pla
ea84  aa      tax
ea85  68      pla
ea86  40      rti
```

Az 1. listán a gép bekapcsolása után végrehajtandó első négy utasítást látjuk. Ezekre korábban már többször hivatkoztam. Itt található a veremmutató kezdőértékének beállítása: a $FCE5 címen lévő utasítás végrehajtása után a mutató a processzorverem utolsó bájtjának címére, $01FF-re mutat, ami azt jelenti, hogy a verem üres. Remélem, senki nem feledkezett meg arról, hogy a veremmutató regiszternek csak az alsó 8 bitjéhez tudunk hozzáférni, a 9. biten mindig 1 van, ezért mindig a $0100 . . . $01FF (256 . . . 511) tartomány címezhető vele.

A 2. listán a C64 IRQ rutinjának az a része látható, amely az A, X és Y regiszter tartalmának a mentése után aszerint határozza meg a megszakítási rutin futásának folytatását, hogy a megszakítást hardveresemény vagy BRK utasítás idézte elő.

A rutinba való belépéskor az utasításszámláló és az állapotregiszter korábbi tartalma már a veremben van. A többi regiszter tartalmát is el kell menteni, mert a megszakított program futásának folytatásához az eredeti regisztertartalmakat helyre kell állítani. Láthatjuk, hogy csak az A regiszter tartalmát tudjuk közvetlenül a verembe tenni, az indexregiszterekét csak az A regiszteren keresztül.

A rutin a regiszterek mentése után az állapotregiszter korábbi állapotát vizsgálja meg. Ehhez a veremmutató tartalmát az X regiszterbe tölti. Ezután arról a címről tölt egy bájtot az A regiszterbe, amely $0104-nek mint báziscímnek és az X indexregiszter tartalmának az összege.

Vizsgáljuk meg, hogy miért éppen $0104 a báziscím. Tudjuk, hogy X-ben a veremmutató aktuális értékének alsó bájtja van, ami a tényleges veremcímtől $0100-zal tér el. Ezt az értéket báziscímnek tekintve, megkaphatnánk a verem következő szabad helyének jelenlegi tartalmát, itt nyilvánvalóan valamilyen — általában a korábbi veremhasználatból származó — „hulladékot" találnánk. A báziscímet eggyel-eggyel növelve, egymás után férhetünk hozzá a verembe korábban letett értékekkel, jelen esetben sorrendben az Y, az X, az A, majd az állapotregiszter mentés előtti tartalmával. Az első hármat a listán látható rutin, az utolsót a BRK utasítás tette a verembe. Az ujjainkon is kiszámolhatjuk, hogy ez utóbbihoz a $0104-es báziscímen keresztül férhetünk hozzá.

A $FF51 címen lévő utasítás végrehajtása előtt az A regiszterben az állapotregiszternek a megszakítás előtti tartalmának megfelelő érték van. A BRK által előidézett megszakítás abban különbözik a hardver IRQ által okozottól, hogy az előbbi az állapotbájt B bitjét 1-re állítja, míg az utóbbi nem. A segédletben (Mikroszámítógép Magazin, 1986/3. A 65XX mikroprocesszor gépi és assembler szintű utasításkészlete) láthatjuk, hogy a B az állapotregiszter 4. bitje, ennek az állapotát kell megvizsgálni. Az AND #$10 utasítás a B eredeti értékével ellentétesen állítja be a Z bit mostani értékét, és ettől függően történik vagy nem történik ugrás $FF58-ra.

A $0314 . . . $0315 című bájtokon a hardver által előidézett megszakítást kezelő rutin címe van elhelyezve, a $0316 . . . $0317 címeken pedig az úgynevezett BRK rutiné. Ezeket a 3-as memórialap elején lévő címeket vektoroknak nevezik, ismertetésükre hamarosan sor kerül.

A *3. listán* a hardver megszakítási rutin befejező utasításait láthatjuk. Az előbbiek ismeretében szükségtelen a magyarázat. A regisztertartalmak visszatöltésének sorrendje fordítottja a mentés sorrendjének.

BARNA LÁSZLÓ

Z80 programok haladóknak Spectrumra és Primóra

5. Festés mintával

Az előző részben láttuk, hogyan lehet zárt alakzatot teljesen befesteni. Tapasztalatainkat és a befestőprogramot felhasználva, már egyszerűen megoldhatjuk a mintával való festés problémáját.

Az *1. listán* a mintás festést megvalósító FILL nevű program látható. Sebessége a PAINT-tal gyakorlatilag megegyező. Inverz zárt alakzat mintás festését is hibátlanul elvégzi.

Az 1. lista mellé be kell még tölteni az előző részben szereplő PAINT rutint, valamint az 1. részben szereplő gyors LDIR-t (LDIRQ). A PAINT rutinban néhány változtatást kell elvégezni, ezek a *2. listán* láthatók. Primo-tulajdonosok a képernyőcím megadására (SCREEN EQU) ügyeljenek.

A program használata

Az indulási pont koordinátáit, akárcsak a PAINT rutinnál, a BC regiszterpárban kell megadni. A minta egy 8×8-as bitminta valahol a memóriában. Címét a DE regiszterpárba kell tölteni. Spectrumon a PROBA rutin itt is a BASIC-ből való hívást támogatja. Ekkor a minta a grafikus "U" karakter képe. (A memória utolsó 8 bájtja.) Primósok írhatnak maguknak hasonló rutint, amely a BASIC CALL utasítás paramétereit átveszi.

A program működése

A program képpuffert használ. Mérete a képernyőével egyezik meg. (Nem pazarlás ez?) A mintás feltöltés menete az *ábrán* követhető nyomon. A bal oldali oszlopban a művelet neve látható, amely egyben a műveletet megvalósító rutinnak a címkéje is. Mellette van a művelet végrehajtása utáni kép-, illetve puffertartalom vázlata.

A STOB rutin bemásolja a képtartalmat a pufferbe. A PAINTB rutin a pufferen festi be az alakzatot. A BXORS rutin bitenkénti kizáró vagy műveletet hajt végre a kép és a puffer között, az eredménybiteket pufferbe írja vissza. Így a pufferben pontosan a feltöltendő alakzat belseje marad. A BANDM rutin a mintát sorban ráilleszti az egész pufferre, és bitenkénti *és* műveletet hajt végre. Az eredménybitek szintén a pufferbe kerülnek. A pufferben az alakzat belseje látható, mintásan. Az ábrán (hibásan) az alakzat kerete is szerepel, másképp nem sikerült szépen megrajzolni. Újra a BXORS rutin fut, a pufferen ekkor megjelenik a kész, feltöltött alakzat. Az EXBS rutin gondoskodik arról, hogy a képernyőre kerüljön a kész ábra.

Mint látható, a műveletek mindvégig a pufferben folynak, a kész kép csak a legvégén kerül a felhasználó csodálkozó szemei elé.

1. lista

```
*L-
*C-
*D+
;*************************
;*                       *
;* FILL-3                *
;* (c)1986.11.29 UHI     *
;*                       *
;*************************

        ORG   $F800
SCREEN  EQU   $4000
BUFFER  EQU   $E000
PATTER  DEFS  2   ;MINTACIM
BUFEND  EQU   BUFFER+$17FF
SCRLEN  EQU   $1800
;=======================
FILL    PUSH  BC
        PUSH  DE
        PUSH  HL
        PUSH  IX
        LD    (PATTER),DE
        CALL  STOB
        CALL  PAINTB
        CALL  BXORS
        CALL  BANDM
        CALL  BXORS
        CALL  BEXS
        POP   IX
        POP   HL
        POP   DE
        POP   BC
        RET
;=======================
STOB    PUSH  BC
        CALL  SBREGS
        CALL  LDIRQ
        POP   BC
        RET
;=======================
BXORS   CALL  SBREGS
BXS1    LD    A,(DE)
        XOR   (HL)
        LD    (DE),A
        INC   DE
        INC   HL
        DEC   BC
        LD    A,B
        OR    C
        JP    NZ,BXS1
        RET
;=======================
BANDM   LD    HL,BUFFER
        LD    IX,(PATTER)
        LD    B,3
BAM1    LD    C,0
BAM2    LD    D,0
BAM3    LD    A,D
        LD    (BAM4+2),A
        LD    A,(HL)
BAM4    AND   (IX+0)
        LD    (HL),A
        INC   H

        INC   D
        BIT   3,D
        JP    Z,BAM3
        LD    A,H
        SUB   8
        LD    H,A
        INC   HL
        DEC   C
        JP    NZ,BAM2
        LD    A,H
        ADD   A,7
        LD    H,A
        DJNZ  BAM1
        RET
;=======================
BEXS    CALL  SBREGS
BES1    LD    A,(DE)
        EX    AF,AF'
        LD    A,(HL)
        LD    (DE),A
        EX    AF,AF'
        LD    (HL),A
        INC   HL
        INC   DE
        DEC   BC
        LD    A,B
        OR    C
        JP    NZ,BES1
        RET
;=======================
SBREGS  LD    HL,SCREEN
        LD    DE,BUFFER
        LD    BC,SCRLEN
        RET

*L-
*C-
*D+
```

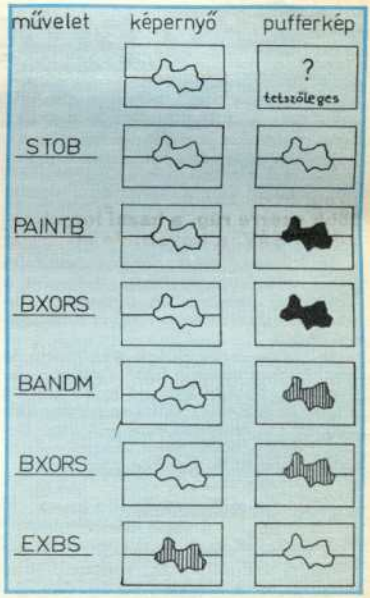
```
;=======================
PAINTB  CALL  TUBINI
        CALL  PIXADB

        CALL  MASKG
        CALL  MODE
        CALL  PUTTUB
PAINT1  CALL  PLINE
        JR    NZ,PAINT1
        RET
;=======================
PIXADB  CALL  PIXADD
        EX    AF,AF'
        XOR   A
        LD    BC,SCREEN
        SBC   HL,BC
        LD    BC,BUFFER
        ADD   HL,BC
        EX    AF,AF'
        RET
;=======================
PROBA   LD    BC,(#5C7D)
        LD    DE,#FFF8
        JP    FILL
```

Érdemes megfigyelni, hogy a program lefutása után a pufferben éppen az eredeti, feltöltetlen ábra van. Ez igen előnyös lehet. Gondoljunk csak arra, hogy a felhasználó valamilyen grafikus szerkesztőprogrammal dolgozik, és éppen be akar tölteni egy általa rajzolt kis alakzatot. Nem veszi észre, hogy egy picurka lyukat hagyott rajta, és mit sem sejtve megnyomja a feltöltést kérő billentyűt. Ekkor a PAINT rutin, természetének megfelelően kibújik a lyukon, és szorgalmasan feltölti az alakzat környezetét is, esetleg az egész képernyőt. Az eredmény: sokkal nagyobb terület lesz mintás, mint ahogyan azt a mesebeli felhasználónk elképzelte. Mit tesz ekkor a szerencsétlen? Nem veszett kárba a többórás rajzolgatása: megnyom egy olyan gombot, amire visszajön a pufferből az eredeti kép, és vidáman kijavíthatja hibáját. Hát ezért nem pazarlás a pufferkép használata. Egyébként egy grafikai szerkesztőprogram más rutinjai is használhatnak pufferképet, ez megegyezhet azzal.

Ezt a FILL rutint a manószerkesztő programban fel fogjuk használni. A következő részben a manószerkesztő további rutinjait mutatom be.

2. lista

UHERKOVICH PÉTER



Verem a Spectrum BASIC-hez

BASIC-ben programozva gyakran kerülünk olyan helyzetbe, hogy feltétlenül verem (stack) használatára volna szükségünk (például rekurzív algoritmusok kódolásakor), de van, amikor a verem használata egyszerűen csak kényelmesebbé teszi a program megírását. A verem fogalmát jól ismerik azok, akik gépi kódban is programoznak. A verem az adattárolásnak egy szervezési formája, amelyet úgy használunk, hogy adatokat írhatunk a verembe, és olvashatunk is belőle. Adatolvasáskor mindig a felső, vagyis az utoljára beírt adatot kapjuk, s az ekkor törlődik a veremből.

A *listán* látható program egy veremtárat valósít meg, a Spectrum BASIC-jéhez illesztve. Adatot írni a PRINT#3 utasítással, olvasni pedig az INPUT#3 utasítással lehet. (Használhatjuk az INKEY$-t is, de nem célszerű!) A program egy CR-rel lezárt szöveget tekint egy adatnak, tehát a verembe írt szövegeket nem „megfordítva" kapjuk vissza. A verembe író PRINT#3 utasítás végén ne legyen pontosvessző, több adatot pedig apposztróffal (') elválasztva írhatunk egy PRINT#3 utasítással.

A vermet az első használat előtt inicializálni kell, hiszen a felsorolt utasítások a nyomtatót működtetik. A verem inicializálása a RANDOMIZE USR 23296 utasítással történjen. Ezután már nem használható a nyomtató, így a program is célszerűen a nyomtató-puffer-területen helyezkedik el. Az inicializálási utasítást egy BASIC programban többször is használhatjuk. Hatása: kiüríti a vermet.

Ha üres veremből próbálunk olvasni, akkor csak CR (enter) karaktert kapunk, hibajelzés nincs. A forrásszöveg 26-os sorától hibajelzést adó utasításokat is be lehet írni: RST 8, utána DEFB hibakód—1. Ekkor RET nem kell.

A program működésében csak egyetlen érdekesebb dolog van: annak a megvalósítása, hogy olvasáskor a szöveg ne forduljon meg. Erre itt most nem térek ki. A forrásszöveg GENS3 assembleren készült.

U. P.

```
 1 ;*********************************
 2 ;*   stack Spectrum basichez     *
 3 ;*********************************
 4         ORG  23296  ;erintre buff
 5 OPEN    LD   HL,WRITE  ;iro rutin
 6         LD   (23749),HL   ;chrout
 7         LD   HL,READ ;olvaso rut.
 8         LD   (23751),HL    ;chrin
 9         LD   HL,(RAMTOP) ;ifo c
10         INC  HL              ; teje
11         LD   (LP),HL  ;"ifo ures"
12         LD   (BP),HL  ;...
13         LD   HL,0     ;...
14         LD   (LL),HL  ;...
15         LD   (BL),HL  ;...
16         RET
17 READ    RES  3,(IY+2)
18         LD   DE,(BL) ;buff. hossz
19         LD   A,D     ;vezet? 0-e?
20         OR   E       ;...
21         JR   NZ,R1   ;nem
22         LD   BC,(LL) ;ifo hossz
23         LD   A,B     ;0-e?
24         OR   C       ;...
25         JR   NZ,R2   ;nem
26         LD   A,13    ;ures: CR
27         SCF          ;kesz!!
28         RET
29 R2      LD   HL,(LP) ;ifopointer
30 R21     DEC  HL      ;CR-t keres
31         DEC  BC
32         INC  DE
33         LD   A,B
34         OR   C
35         JR   Z,R3    ;kiurult!
36         DEC  HL
37         LD   A,(HL)
38         INC  HL
39         CP   13      ;CR?
40         JR   NZ,R21  ;nem
41 R3      LD   (LP),HL ;uj pointer
42         LD   (LL),BC ;uj hossz
43         LD   (BP),HL ;buffer cim
44 R1      LD   HL,(BP) ;buffer olv.
45         LD   A,(HL)  ;kiolv. char
46         INC  HL      ;kpoint. ter
47         LD   (BP),HL ;uj bufrpint
48         DEC  DE      ;hossz csokk
49         LD   (BL),DE ;uj hossz
50         SCF          ;"van chor"
51         RET
52 WRITE   LD   HL,(BP)
53         LD   DE,(LP)
54         OR   A       ;cf torles
55         SBC  HL,DE   ;BP=LP?
56         JR   Z,W1    ;igen
57 W3      LD   (DE),A  ;char bele
58         INC  DE      ;pointer nov
59         LD   (LP),DE ;uj tetpointer
60         LD   HL,(LL) ;ifofohossz..
61         INC  HL      ;..noveles
62         LD   (LL),HL
63         RET
64 W1      LD   HL,(BP)
65         LD   (CH),A  ;char mentes
66         LD   BC,(BL)
67         LD   DE,(UDG)    ;ham cim
68         ADD  HL,BC
69         PUSH HL      ;buffer vege
70         SBC  HL,DE   ;elerte"
71         POP  DE      ;BP+BL
72         JP   Z,#1F15 ;OM error
73         LD   H,B
74         LD   L,E
75         LD   A,B
76         OR   C
77         JR   Z,W4
78         DEC  HL
79         LDDR         ;helyesmolas
80 W4      LD   DE,(LP)
81         LD   HL,(BP)
82         INC  HL
83         LD   (BP),HL
84         LD   A,(CH)
85         JR   W3
86 LP      DEFS 2   ;ifo-pointer
87 LL      DEFS 2   ;ifo-tenyht
88 BP      DEFS 2   ;buffer-pointer
89 BL      DEFS 2   ;buffer-tenyht
90 CH      DEFS 1   ;work
91 UDG     EQU  23675  ;felso heter
92 RAMTOP  EQU  23730  ;ifo kezdet
```

A Spectrum másolói, másolók spektruma

Miközben folyik a vita a szoftverlopásokról, a hazai Spectrum-tulajdonosok programjaik többségét ha akarnák sem tudnák másképpen megszerezni, csak cserebere, azaz másolás útján. A vásárolható szoftver nemcsak drága, de választéka is elképesztően szegényes. Míg a Spectrumra készült programok száma a nagyvilágban több ezerre rúg, a hazai forgalmazók (a BÁV, a Novotrade stb.) alig tucatnyit tudnak ebből kínálni. Ilyen körülmények között a másolóprogramok alkalmazása szükséges rossz, nem pedig erkölcsi kérdés.

A hazai Spectrum-amatőrök körében többtucatnyi másolóprogram van forgalomban. Érdemes is belőlük minél többre szert tenni, mert alapszabály, hogy amit az egyikkel nem tudunk lemásolni, azt lemásolhatjuk egy másikkal; azaz nincs másolhatatlan program, csak esetleg még nem szereztük meg a megfelelő másolóprogramot.

Ismeretes, hogy a Spectrum egyik nagy előnye az állományok (ismertebb nevükön: fájlok) betöltésének és kimentésének rendkívüli egyszerűsége. A megfelelően „ragozott" LOAD, SAVE parancsokkal minden szabványos állomány (BASIC, gépi kód, képernyő, adattömb) könnyedén kezelhető. Ilyen állományokat kis ügyességgel minden nehézség nélkül közvetlenül másolhatunk, még másolóprogram sem kell hozzá. A rafináltan elkészített, különböző trükkökkel titkosított állományokat azonban már csak speciális másolóprogramokkal vehetjük át saját programkönyvtárunkba.

A másolóprogramok megítéléséhez tudni kell, hogy egy szabványos Spectrum-állomány két részből: a fejből és a törzsből áll. Tulajdonképpen mind a kettő egy-egy fejnélküli állomány, és a ROM-program egy speciális jelzőbájt — az OMNICOPY terminológiájával markbájt — alapján különbözteti meg őket: a fejnél ez 0, a törzsnél 256. (Jó lenne, ha valaki közölné, hogy hol található ez a bájt!)

A fej 17 bájton tartalmazza a törzs adatait; leírásuk több magyar nyelvű publikációban megtalálható. Ezeket az adatokat minden valamirevaló másolóprogram kiírja.

Melyiket és miért?

A másolóprogramokat két osztályba sorolhatjuk: a *szimpla másolók* egyszerre csak egy fájl másolását teszik lehetővé, a *multimásolókkal* pedig egyszerre több állomány is másolható. Mivel egy nagyobb program általában több állomány fűzére, a szimpla másolókkal elég kényelmetlen dolgozni. A továbbiakban a multimásolókról lesz szó.

Egy multimásoló megítélésénél a következő szempontokat mérlegeljük:

— mekkora a szabad tárterülete,
— hány darab állomány tölthető be egyszerre,
— hogyan törölhetők a feleslegesen felvett állományok,
— milyen különleges (fejnélküli, hamisfejes, turbósított stb.) állományok másolhatók vele,
— milyen fejadatokat tud kiírni.

Egy jól megírt multimásoló szabad tárterülete kb. 40—42 k. Az ennél nagyobb állományok, a megafájlok speciális másolással másolhatók.

A legtöbb másolóprogram a képernyő felső vagy alsó harmadában helyezkedik el, és vagy látható, vagy elfedik az attributumok. Ebbe a harmadba természetesen semmit sem lehet írni, vagyis a képernyőn egyszerre legfeljebb 10—15 állomány adatai jeleníthetők meg. A legtöbb másolónál ez behatárolja a betölthető állományok számát. Vannak azonban kivételek is: például az OMNICOPY mozgó állománylistát készít és jelenít meg, ezért a betölthető állományok számának csak a szabad tárkapacitás szab határt.

Egyes másolóknál — például az FM3-nál — egyszerre törölhető valamennyi addig betöltött állomány. Ez néha nagyon kényelmetlen. A jobb másolók munkaállomány-jelzői (a fájlkurzorok) állíthatók — az FM3-nál csak SAVE módban —, és a törlés csak a munkaállománytól kezdődik. Ilyenek például a Zotyocopy+ és a COPY—COPY. A legtökéletesebb, ha csak a munkaállomány törlődik, mint például az OMNICOPY-nál.

Egy másoló annál jobb, minél többféle különleges állományt tud kezelni. Nem ismerek olyan multimásolót, amely nem tud fejnélküli állományt másolni; az ilyen nem is érne sokat. Viszont például a turbósított állományok kezelése már extra szolgáltatásnak számít. A legtöbb fejtörést a hamisfejes állományok okozzák, melyek az állomány típusát, hosszát és betöltési címét illetően valótlan adatok tartalmaznak. A beolvasásukra képes másolók sajnos csak a legdurvább szalaghibát jelzik hibaüzenettel, és így ismeretlen állománynál sokszor nem tudhatjuk, hogy a fej hamis-e, vagy a beolvasás volt-e hibás. Általában eredményt hoz a megismételt betöltés: ha ugyanis a valódi hossz másodszor nem ugyanaz, mint először, akkor biztos, hogy rossz volt a beolvasás. Az ellenkező eset sajnos semmit nem bizonyít, mert az azonos hosszt egy makacs szalaghiba is eredményezhette.

Itt említem meg, hogy minden tisztességes másoló kiírja a fejadatokat és a tényleges betöltési hosszt. Egyes másolók tízes számrendszert használnak, mások tizenhatost. Általában a tízes alak kényelmesebb, de éppen a gyanús állományok tesztelésénél hasznos lehet a tizenhatos alak is. Az elegánsan megírt gépi kódú állományok hossza ugyanis gyakran kerek szám, de a kerekség érthető hexadecimálisan is. Például a képernyőállomány hossza 6912 bájt, ami hexadecimálisan 1B00. Ha tehát egy fejnélküli vagy hamisfejes kétes állomány hossza kerek szám (tízesben vagy tizenhatosban), akkor már nem is olyan gyanús. De minden gyanú felett csak a hibátlanul futó programok állnak.

A jó másolóprogramtól extra szolgáltatásokat is várhatunk, mint például a szabad tárterület nagyságának kijelzése, a fej és a törzs külön állománykénti kezelése (ez akkor hasznos, ha az állományok egyenként törölhetők), a fejnélküli állomány megfejelése, az autoRUN megszüntetése, a másolat ellenőrzése (VERIFY), a fej módosítása, a fej titkosítása, megafájlmásolás, katalógus készítése, monitor üzemmód stb.

A másolóprogramokkal általában nem nehéz boldogulni, csak némi leleményességre és nagy türelemre van szükség. Legtöbbjük „önmagyarázó", sőt sokat meg is „magyarítottak". Mégis úgy gondolom, nem hiábavaló, ha megosztom tapasztalataimat az olvasókkal az általam előszeretettel használt négy multimásoló sajátosságairól. Három közülük — a Zotyocopy, a COPIER FM3 és az OMNICOPY—2 — Spectrum-körökben valószínűleg elterjedt, de a negyedik, a COPY—COPY talán kevésbé ismert.

A Zotyocopy

Hibátlanul felvett, hibátlanul működő, nem turbósított és nem meghamisított fejű állományok másolására a Zotyocopy újabb változatait, a Zotyocopy+-t és a Zotyocopy#-t ajánlom. A kettő között a megafájlmásolásban van különbség. Az előbbivel tetszőleges számú megafájlról lehet egy-

egy másolatot készíteni, az utóbbival egyetlen állomány másolható le több példányban. Egyébként a két változat azonos. A Zotyo legnagyobb előnye, hogy tiszta, szép másolatokat készít. A kék-piros csíkokat okozó betöltő részek hossza majdnem akkora, mint a közvetlenül SAVE-vel kimentett állományoknál. A fej és a törzs között érzékelhető szünetet tart, az egyes állományok között pedig ennél valamivel hosszabbat, ami lehetővé teszi össze nem tartozó állományok egy töltéssel való másolatát. A SAVE (S), VERIFY (V) és FORGET (F — törlés) üzemmódokban a fájlkurzor a D (Down — le), illetve az U (Up — fel) gombokkal mozgatható. A SAVE és a VERIFY kapcsolható kézzel (Hand), amikor csak a kijelölt munkaállományokkal dolgozunk, vagy automatikusan (Auto), amikor a munkaállománytól az állománysor végéig történik a feldolgozás. A törlés a munkaállományra és az alatta felsorakozó állományokra vonatkozik.

AutoSAVE esetén a másolás az Enter lenyomása után néhány másodperccel kezdődik, HandSAVE esetén viszont azonnal.

A megafájlmásoló a Capsshift + M gombok megnyomásával indítható. Természetesen az alapüzemmódba való visszatérés nem lehetséges, ami egyébként minden megamásolóra igaz. A Zotyo, ellentétben például a TICY (BTICY) szimpla megamásolóval, csak fejnélküli megafájl másolására alkalmas. A fej elválasztását például az OMNICOPY-val oldhatjuk meg, amely ugyan maga is képes megamásolásra, de túlságosan rövid betöltő résszel.

Tapasztalatom szerint a bizonytalan felvételeket az OMNICOPY általában jobban „megeszi", mint a Zotyo, bár előfordult már az ellenkezője is. Olyan programot, ahol a fejben közölt állományhossz nem egyezik a ténylegessel, a Zotyo általában nem fogad el. Ezeknél az FM3 vagy az OMNICOPY segíthet.

Ha a Zotyo betöltési hibát vagy memória-túlcsordulást jelezve leáll, az L (LOAD) megnyomása újraindítja, miközben a hibás munkaállomány törlődik. Break vagy más üzemmód nem törli le a hibás állományt, de nem is kezeli.

A COPIER FM3

A COPIER FM3 nem ellenőriz, a fejadatokat hexadecimálisan írja ki (léteznek decimális variánsok is), nem jelzi ki a szabad tárkapacitást, a betöltött állományokat csak egyszerre tudja kitörölni. Ezek a hátrányai. Előnye, hogy hibás és hamisfejes állományokat is elfogad; csillaggal jelöli őket a képernyő utolsó oszlopában. A hibás állományok nem mindig hibásan másolódnak ki!

Megfelelő türelemmel az FM3-at programmentés eszközeként is használhatjuk. A másolás indításakor hangjelzést ad, ami előnyös lehet a visszakereséskor. Az AutoSAVE módban a fájlkurzortól kezdi a kimentést. Az állományok betöltő részei valamivel rövidebbek, mint a Zotyonál, de azért elég hosszúak. A fej és a törzs közötti, illetve az automatikusan felvett állományok közötti szünetek hossza kielégítő. A fájlkurzort az S billentyű mozgatja.

Az OMNICOPY—2

Az egyik legkiválóbb multimásoló az OMNICOPY—2. Mindössze két hátránya van a Zotyóval és az FM3-mal szemben: egyrészt nincs lehetőség egyenkénti másolásra, másrészt túlságosan takarékosan bánik az idővel és a mágnesszalaggal; igen rövidek a betöltő részek és fej a törzzsel, állomány az állománnyal teljesen összefolyik. Annál több azonban az előnye! Külön állományként kezeli a fejet és a törzset, a munkaállomány egyenként az állománysoron belül is törölhető (Q), a fejnélküli állományok megfejelhetők (H), átállítható dupla sebességű turbó üzemre (D), megszüntethető az autoRUN (R), mega- és maxiállományokat is másol (M és N), titkosítja a fejet (Headerer Pitch on — P), megszámolja az állomány bájtjait (C), a fejadatok között közli a markbájt tartalmát és a BASIC program változómezejének hosszát.

Ha a Break megnyomásával leállítjuk az OMNICOPY-t és az L gombot nyomjuk meg, a LOAD üzemmód a tár teljes kitisztításával kezdődik. Ha folytatni akarjuk az állománysort, akkor a B gombot kell megnyomni, majd az Entert. Az OMNICOPY így is LOAD módba kerül, de nem törli a munkatárat.

A COPY—COPY

A végére hagytam az Eniac Software által forgalmazott COPY—COPY multimásolót. A feliratok alapján valószínűleg lengyel termék. Rendkívül sokoldalú másoló, az OMNICOPY-val együtt használva a kódtörés és a programtitkosítás hatásos eszköze. Különböző funkciói egy-egy opcióbillentyű, majd az Enter lenyomásával hozhatók működésbe.

(J) LOAD <n> Állomány beolvasása az n-edik helyre, n legfeljebb az utolsó sorszámnál egygyel nagyobb és 15-nél kisebb lehet. Ha n nem szerepel, akkor a betöltés a sorban következő helyre történik. Az n-edik helytől kezdve a már betöltött állományok törlődnek.

(S) SAVE <n> Állománykimentés az n-edik helytől végig. Egyenkénti (Hand) kimentésre nincs mód, n hiánya esetén a kimentést az első állománytól kezdi.

(V) VERIFY <n> Ellenőrzés. Az n szerepe ugyanaz, mint az S-nél.

(Z) COPY Megafájlmásoló

(L) LET n = név Az n-edik állomány átnevezése névre. A név-x sztringet nem szabad idézőjelbe tenni, az is része lesz az új névnek. Csak az első tíz karaktert használja fel. Ha nem adunk meg nevet, összezavarodhat a rendszer!

(C) CAT Szalagkatalógus

Monitorparancsok

(K) LIST <a> (0 ≤ a ≤ 65535). Négyoszlopos monitorlista az a címtől kezdve. A négy oszlop a következő: 1. cím, 2. a címen kezdődő kétbájtos szó, 3. a címzett bájt tartalma (dd), 4. CHR$ dd. A nem opcionált billentyűk lapoznak, az opcionáltak kiadják saját parancsukat.

(O) POKE < a,dd > Az a címen elhelyezi a dd bájtot. Egylapnyi monitorlistát ad akkor is, ha dd-t nem adunk meg.

(U) USR <a> Az a címtől elindít egy gépi kódú programot. Nagyon óvatosan kell vele bánni, mert igen könnyen elszállhat a rendszerünk.

(Y) RETURN Visszatérés a BASIC-hez, a másoló törlése a tárból.

A rosszul beütött, nem kívánt opciók a DELETE segítségével törölhetők. A folyamatban levő funkciókat a Breakkel lehet leállítani. Enter lenyomása esetén némely funkció (VERIFY, SAVE stb.) ismét jelentkezik.

A maximálisan 15 db betölthető program feladatai a következő szavak alatt jelennek meg:

nazwe — sorszám (n) és név

typ — tipus (P, B stb.)

dlug — a program hossza. Ezt az adatot a tényleges hossz felülírja

start — autoRUN sorszám, illetve betöltési cím

prog — ennek az információnak az értelmét nem sikerült megfejtenem.

Az utolsónak betöltött program neve alatt WOLNE inverz felirat látható, melette a „dlug" oszlopban a még szabad tárkapacitás olvasható.

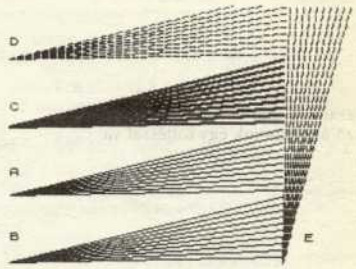
A betöltő részek elég hosszúak, de az egyszerre felvett állományok között érzékelhető szünet nincs, ezért csak összetartozó állományokat célszerű egy menetben másolni. A másolás befejeztét búgó hang jelzi, ami jól elválasztja egymástól a felvételeket.

DR. NAGY ANDRÁS

Számítógépes grafika Pascalban

EGYENES

ZX-Spectrum



A ZX-Spectrumra készült Hisoft Pascal kitűnő eszköz olyan programok írására, amelyeknél a gyorsaság különösen fontos szempont. Ilyen területe a számítástechnikának a grafika is. Úgy hisszük, hogy az alábbi sorokat — és majd a következő számainkban a folytatást jelentő cikkeket is — tehát azok az olvasók is hasznosíthatják, akik valamilyen oknál fogva nem akarnak Pascalban programozni, viszont szeretnének megismerkedni a számítógépes grafikában használatos módszerekkel.

Néhány olyan számítógépes grafikai alapfeladat Pascalban írt megoldására mutatunk példát, amelyek megvalósítására a BASIC interpreterek lassúságuk miatt gyakorlatilag alkalmatlanok. A sebesség mellett a Pascal programok előnye, hogy a forrásnyelvű szöveg strukturáltsága következtében jól olvashatók és igen alkalmasak az algoritmusok tanulmányozására.

Sajnos a Hisoft Pascal nem ismer grafikai utasításokat, eljárásokat vagy függvényeket, szerencsére azonban a BASIC-hez hasonlóan lehetőséget ad gépi kódú eljárások hívására. Így a hiányzó és alapvető grafikai eszközöket magunk is megírhatjuk. Nos, rajzolás elvben is csak akkor lehetséges a képernyőre, ha legalább egy pont megjelenítésére szolgáló eljárásunk van.

Az *1. lista* néhány grafikai eljárást tartalmaz. Ezek közül az első, a PLOT azonosítójú egy X, Y koordinátájú pont felrajzolására szolgál. A koordinátarendszer kezdőpontját a BASIC PLOT utasításához hasonlóan a képernyő bal alsó sarkába helyeztük. A többletet az adja, hogy a teljes képernyőre rajzolhatunk (az alsó két karaktersorba is), tehát a képernyő mérete 256×192 pont. Még egy előnye van az itt közölt PLOT eljárásnak a BASIC PLOT utasításával szemben: ha X vagy Y értéke nagyobb, mint a fenti határ, vagy negatív, ha tehát képernyőn kívülre esik, nem áll le a program, hanem egyszerűen nem hajtja végre az utasítást.

A gépi kódú rutin a COORDS rendszerváltozóba (23672=507DH) tölti az X és Y koordinátát, majd az A regiszterbe 191-et — az Y irányban lehetséges pontok legnagyobb számát — tölti, és meghívja a képelem címe (PIXEL-ADDRESS) ROM rutint. Ezt követően hívja a PLOT szubrutint a 22ACH és a 22ECH címen. (Így végül is a ROM rajzoló eljárást használjuk a Pascalban is.) Mivel az egész kis gépi kódú programrész relokálható, egy INLINE-ba ágyazhattuk.

A LINE eljárás a ROM egyenesgenerátorát használja a BASIC koordinátarendszere szerint. Csak azért mutatjuk be, hogy össze lehessen hasonlítani a Pascal egyenesgenerátorával. Akit az algoritmus működése érdekel, megtalálhatja az ismertetését a Hisoft Pascal leírásában.

A PixelLine azonosítójú eljárás az előbbiekben bemutatott PLOT segítségével egyenes szakaszt rajzol. Az egyenes rajzolásánál az a probléma, hogy egyenletének a képernyő raszterbeosztásának megfelelő pontjaiból olyan alakzatot hozzunk létre, amelyet az emberi szem egyenessé egészít ki. Könnyen belátható, hogy raszteres grafikáknál pontos egyenes csak kivételes esetekben rajzolható (például ha a párhuzamos valamely koordinátatengellyel vagy a szögfelezőjükkel párhuzamos). Egyéb esetekben csak az biztosítható, hogy a felrajzolt pontsor az elméletileg pontos egyenes közvetlen környezetében helyezkedjen el.

A programunkban azt az eljárást valósítjuk meg, amelyet Bresenham fejlesztett ki. Az eljárás az X1, Y1 és X2, Y2 koordinátájú pontok közötti egyenes szakaszt jeleníti meg. A végrehajtandó algoritmus 45 fok hajlásszög alatti és feletti egyeneseknél eltérő. Ha az egyenes meredekségének abszolút értéke kicsi, további két eset lehetséges: az egyenes jobbra vagy balra mutat. Mindkét esetben az egyenest úgy rajzoljuk fel, hogy X irányban az egyenes pontjai a lehető legsűrűbben helyezkedjenek el, tehát raszterosztásonként léptetjük X értékét. Az X növekmény, melyet a program A-val jelöl, 1 vagy −1 attól függően, hogy a szakasz az első ponttól jobb vagy bal irányba fut.

Az Y irányú növekményt a program annak megfelelően állítja be, hogy az egyenes az első ponttól fel vagy le mutat. A B növekmény ennek megfelelően 1 vagy −1. Ennyi előkészület után már hozzákezdhetnénk az egyenes rajzolásához, csak még azt kellene eldönteni, hogy az egyenletes X irányú lépdelés során mikor kell az Y irányú lépést végrehajtani. Ehhez a D változóban tárolt ún. döntési függvényt használjuk, amely azt tartja számon, hogy a pontos egyenes az éppen felrajzolandó pont alatt vagy felett haladna-e el.

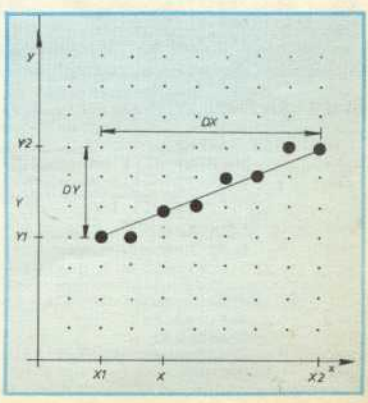
Az egyenes rajzolása mindig valóságos képpontból indul. Ez azért van így, mert X1 és Y1 integer paraméterek. Az egyenes elméleti egyenlete a vázlat szerint:

$$Ye = \frac{DY}{DX} * (X - X1) + Y1$$

Ye általában nem egész szám, ezért el kell döntenünk, hogy az előző pont integer Y értékét az új pont esetén növeljük-e vagy sem. Úgy célszerű eljárnunk, hogy ha az elméleti egyenes az eredeti Y értékhez közelebb esik, Y-t változatlanul hagyjuk, ha messzebb, Y-t eggyel növeljük. Ennek a kritériumnak az alábbi vizsgálat felel meg: Ha $Ye - Y >= 0{,}5$, akkor Y legyen $Y + 1$ vagy ha

$Ye - Y - 0{,}5 >= 0$, akkor Y legyen $Y + 1$.

A fenti vizsgálat könnyen elvégezhető az



ún. hibafüggvény (E) vizsgálatával, amely az előbbiek szerint

$$E = \frac{DY}{DX} * (X - X1) + Y1 - Y - 0{,}5$$

összefüggésből számítható. A képlet használata azonban kényelmetlen, mivel lebegőpontos (valós) számokkal dolgozik, és az ezekkel végzett műveletek időigényesek. Számunkra a gyorsaság igen fontos, hiszen ezt a számítást minden pontra el kell végezni. Egy kis ügyességgel elkerülhetjük a lebegőpontos műveleteket. Ha mindkét oldalt beszorozzuk 2*DX-szel, az így kapott

D = 2*DY*(X − X1) − 2*DX*(Y − Y1) − − DX

1. lista

```
  10 {$L-}
  20 {1:LINE fájlenevvel kimentett include rutinok}
  30
  40 PROCEDURE PLOT(X,Y:INTEGER);
  50 {Pont rajzolo eljaras. A ROM rutinjat
  51 hasznalja. Csak akkor rajzol, ha a pont
  60 a 0(=x(=255 es 0(=y(=191 meretu
  61 kepernyore esik.}
  70 BEGIN
  80   IF (X)=0)AND(X(=255)AND(Y)=0)AND(Y(=191) THEN
  90   INLINE(#FD,#21,#3A,#5C, {LD IY,#5C3A)
 100   #DD,#46,2,              {LD B,(IX+2)}
 110   #DD,#4E,4,              {LD C,(IX+4)}
 120   #ED,#43,#7D,#5C,        {LD (#5C7D),BC)
 130   #3E,#BF,                {LD A,191}
 140   #CD,#AC,#22,            {CALL #22AC)
 150   #CD,#EC,#22             {CALL #22EC))
 160 END; {PLOT}
 170
 180 PROCEDURE LINE(DX,DY:INTEGER);
 190 {Az alabbi ket eljaras a Spectrum ROM
 191 egyenes rajzolo rutinjat hasznalja
 200 HISOFT alapjan}
 210 VAR SGNX,SGNY:INTEGER;
 220
 230 PROCEDURE LINE1(X,Y,SX,SY:INTEGER);
 240 BEGIN
 250  INLINE(#FD,#21,#3A,#5C,
 260         #DD,#56,2,#DD,#5E,4,
 270         #DD,#46,6,#DD,#4E,8,
 280         #CD,#BA,#24)
 290 END; {LINE1}
 300
 310 BEGIN
 320  WRITE(CHR(21),CHR(0));
 330  IF DX(0 THEN SGNX:=-1 ELSE SGNX:=1;
 340  IF DY(0 THEN SGNY:=-1 ELSE SGNY:=1;
 350  LINE1(ABS(DX),ABS(DY),SGNX,SGNY)
 360 END; {LINE}
 370
 380 PROCEDURE PixelLine(X1,Y1,X2,Y2:INTEGER);
 390 {Ez az eljaras viszont csak a pont
 391 rajzolo algoritmust hasznalja a ROM
 400 programjabol, az egyenes pontjainak
 401 koordinatait a Bresenham modszerrel
 410 Pascalban szamitja ki}
 420 VAR X,Y,Z,A,B,DX,DY,D,DELTAP,DELTAQ:INTEGER;
 430 BEGIN
 440   DX:=ABS(X2-X1);
 450   DY:=ABS(Y2-Y1);
 460   IF DY(=DX THEN {Az iranytangens (=1}
 470   BEGIN
 480     X:=X1; {X kezdoerteke}
 490     Y:=Y1; {Y kezdoerteke}
 500     Z:=X2; {X iranyt figyeli}
 510     {Beallitja az X iranyu novekmenyt}
 520     IF X1(=X2
 530       THEN A:=1 {X novekszik}
 540       ELSE A:=-1;{X csokken}
 550     {Beallitja az Y iranyu novekmenyt}
 560     IF Y1(=Y2
 570       THEN B:=1   {Y novekszik}
 580       ELSE B:=-1; {Y csokken}
 590     {Inicializalja a dontesi fuggvenyt
 591     es deltait}
 600     DELTAP:=DY+DY;
 610     D:=DELTAP-DX;
 620     DELTAQ:=D-DX;
 630     {Meghatarozza a pont helyet es kinyomtatja}
 640     PLOT(X,Y); {Elso pont}
 650     WHILE X<>Z DO
 660     BEGIN
 670       X:=X+A;
 680       IF D(0
 690         THEN D:=D+DELTAP
 700         ELSE
 710         BEGIN
 720           Y:=Y+B;
 730           D:=D+DELTAQ
 740         END;
 750       PLOT(X,Y);
 760     END {WHILE}
 770   END  {Ha DY(=DX}
 780   ELSE {Ha DY)DX}
 790   BEGIN
 800     Y:=Y1; {Y kezdoerteke}
 810     X:=X1; {X kezdoerteke}
 820     Z:=Y2; {Y iranyt figyeli}
 830   {Y novekmeny beallitasa}
 840     IF Y1(=Y2
 850       THEN A:=1   {Y novekszik}
 860       ELSE A:=-1; {Y csokken}
 870   {Beallitja X novekmenyt}
 880     IF X1(=X2
 890       THEN B:=1   {X novekszik}
 900       ELSE B:=-1; {X csokken}
 910     {Inicializalja a dontesi fuggvenyt
 911     es deltait}
 920     DELTAP:=DX+DX;
 930     D:=DELTAP-DY;
 940     DELTAQ:=D-DY;
 950     {Meghatarozza a pont helyet es
 951      kinyomtatja}
 960     PLOT(X,Y); {Elso pont}
 970     WHILE Y<>Z DO
 980     BEGIN
 990       Y:=Y+A;
1000       IF D(0
1010         THEN D:=D+DELTAP
1020         ELSE
1030         BEGIN
1040           X:=X+B;
1050           D:=D+DELTAQ
1060         END; {ELSE}
1070       PLOT(X,Y);
1080     END {WHILE}
1090   END {ELSE}
1100 END; {PixelLine}
1110
1120 PROCEDURE FatLine(X0,Y0,X,Y,H:INTEGER);
1130 {Vastag egyenes rajzolasa}
1140 VAR I,J:INTEGER;
1150 BEGIN
1160   FOR I:=1 TO H DO
1170   FOR J:=1 TO H DO
1180   PixelLine(X0+I-1,Y0+J-1,X+I-1,Y+J-1)
1190 END; {FatLine}
1200
1210 PROCEDURE DotLine(X0,Y0,X,Y:INTEGER);
1220 {Szaggatott vonal rajzolasa}
1230 CONST H=4;D=6;
1240 VAR I,N,XX0,XX:INTEGER;
1250     T,YY0,YY:REAL;
1260 BEGIN
1270   IF ABS(X-X0))0 THEN
1280     T:=(Y-Y0)/(X-X0)
1290   ELSE T:=2000;
1300   IF T(1 THEN
1310   BEGIN
1320     N:=ROUND((X-X0)/D);
1330     XX0:=X0;
1340     XX:=X0+H;
1350     FOR I:=0 TO N DO
```

```
1360      BEGIN
1370        YY0:=T*XX0+Y0;
1380        YY:=T*XX+Y0;
1390        PixelLine(XX0,ROUND(YY0),XX,
1391        ROUND(YY));
1400        XX0:=XX0+D;
1410        XX:=XX+D;
1420      END
1430    END ELSE
1440    BEGIN
1450      T:=1/T;
1460      N:=ROUND((Y-Y0)/D);
1470      XX0:=Y0;
1480      XX:=Y0+H;
1490      FOR I:=0 TO N DO
1500      BEGIN
1510        YY0:=T*XX0+X0;
1520        YY:=T*XX+X0;
1530        PixelLine(ROUND(YY0),XX0,
1531        ROUND(YY),XX);
1540        XX0:=XX0+D;
1550        XX:=XX+D
1560      END
1570    END
1580 END; {DotLine}
L
```

2. lista

ún. döntési függvény vizsgálatát kell csak végrehajtanunk, mely érték egész szám. A PixelLine eljárás 600-760-as soraiban ezt az algoritmust programoztuk azzal a módosítással, hogy a kijelölt szorzásokat is sorozatos összeadással helyettesítettük.

Az eljárás második része a fenti algoritmust ismétli meg arra az esetre, ha az egyenes iránytangense a 45 fokos egyeneséhez képest nagyobb. Ennek működése az előzőek szerint kézenfekvő. Most az Y-t léptetjük egyenletesen, és a döntési függvénnyel azt vizsgáljuk, mikor kell az X-en változtatni.

Az eddigiek alapján egy sor egyenesekkel kapcsolatos feladatot oldhatunk meg. Az *1. lista* FatLine eljárásával tetszőleges vastagságú egyeneseket jeleníthetünk meg. Az eljárás paraméterei megegyeznek a PixelLine-éval, de kiegészülnek még egy H integerérték szerint hívott paraméterrel, amely a vonal vastagságát adja meg képponttávolság egységben.

A DotLine eljárás viszont szaggatott vonalat rajzol. Az eljárásba nem paraméterként, hanem állandóként definiáltuk, hogy az egyenes négy képpontnyi egyenes szakaszokból (H) és ezeket elválasztó két képpontnyi szünetekből áll, amely mintázat tehát hat képpontonként ismétlődik.

Az ismertetett eljárásokhoz készített demonstrációs program a *2. listán* látható. Ezt többféleképpen is kipróbálhatjuk. Célszerű először a Pascal szövegszerkesztőjével az 1. listát elkészíteni, majd elmenteni magnókazettára vagy Microdrive cartridge-ra. (A listák a lemezmeghajtós változatot mutatják be.) Itt a fájlt P1,9999,1:LINE paranccsal mentjük ki. Ha ezután elkészítjük a 2. lista programját és azt C paranccsal lefordítjuk, a fordítás a 70-es sornál INCLUDE parancsot talál, amelynek hatására a képernyőre kiírja a fájlnevet, beolvassa a meghajtóból a LINE nevű szövegfájlt és befordítja a főprogramba. Ilyenkor azt tapasztalhatjuk, hogy fordítás közben a meghajtó többször beindul, leáll. Ez nem hiba, mivel mindig csak egy puffernyit olvas be a gép, ezt lefordítja, majd ha kifogy a szövegből, újabb adagot olvas be és így tovább.

A másik lehetőség a következő. Ha a demonstrációs program bebillentyűzése során a 60-as sorig jutunk, G„1:LINE paranccsal hívjuk be a meghajtóról az egyenesrajzoló eljárásaink forrásszövegét. Ez a művelet a BASIC-kel ellentétben nem írja felül a memóriában levő programot, hanem hozzáfűzi. Egyidejűleg a sorokat egyesével át is számozza, de ez az N paranccsal megváltoztatható. A beolvasás után a főprogram bebillentyűzését a 2. lista 90-es soránál folytassuk. Természetesen ez a sor az összeszerkesztett listában nem a 90. lesz.

Ennek a második módszernek az az előnye, hogy a fordításhoz a lemezmeghajtót nem kell használnunk, ami a munka lényeges gyorsítását jelenti. Hátránya viszont, hogy ezzel a módszerrel csak kisebb programok fordíthatók. Azt tehát, hogy melyik módszer a célszerűbb, a feladat jellege dönti el.

Kazettás magnetofonnal a fenti eljárás kissé módosul. A fájlnév ebben az esetben LINE lesz. Két fájlt kell készítenünk az eljárásokról. Az egyiket P1,9999,LINE paranccsal hajtjuk végre. Ezt lehet majd a fent ismertetett második módszer szerint beszerkeszteni a főprogram szövegébe. Az INCLUDE módszerhez a fájlt a W1,9999,LINE paranccsal kell a kazettára menteni. Ekkor a Pascal a fájlt 128 bájtos rekordok formájában viszi ki szalagra, és közöttük elegendő szüneteket hagy ahhoz, hogy visszaolvasás közben a fordítást a fordítóprogram elvégezhesse.

Az eljárások külön fájlokban való tárolása azért célszerű, mert ezekből egy új program gyorsan összeszerkeszthető. Az ismertetett eljárásokat úgy írtuk meg, hogy globális változókat nem tartalmaznak, így az új programnál csak a paraméterek kitöltésére kell ügyelnünk. Gyakorlott Pascal programozó az idők folyamán egész kis eljárás- és funkciókönyvtárat gyűjthet össze, amelynek segítségével a Pascalból ízlésének megfelelő, mondhatni testre szabott nyelvet alakíthat ki.

Végül a bemutató programról annyit, hogy a grafikai példa A egyenesseregét a LINE eljárással, tehát az eredeti ROM rutinnal, a B-vel jelölt egyeneseket pedig a PixelLine eljárás segítségével rajzoltuk fel. Mivel a két egyenessereg egyes elemei egymással párhuzamosak, jól látható, hogy ugyanolyan algoritmusra épül mindkét eljárás.

DR. KABOLDY PÉTER

AZ ATARI-PÁLYÁZAT NYERTESEI

A KISZ KB Középiskolai és Szakmunkástanuló Tanácsa (KSZT) a Skála Coop számítástechnikai leányvállalatával, a Computer—S-sel közösen „ATARI-PÁLYÁZAT"-ot hirdetett (lásd magazinunk 1987/1. számában).

A KSZT a beérkezett pályázatokat katalogizálta és titkosította, majd bírálatra átadta a Skálának.

A nyertesek és díjaik:

1. **Reith József** *(Budapest, Városmajor u. 66.)*
   ATARI 800 XL számítógép teljes kiépítéssel, egyéves használatra
2. **ifj. Kékkői Zoltán** *(Villány, Táncsics Mihály u. 13.)*
   ATARI 800 XL számítógép teljes kiépítéssel, féléves használatra
3. **Kovács Tibor** *(Szeghalom, Kinizsi u. 37.)*
   ATARI-program kazettán
4. **ifj. Brosig János** *(Szombathely, Bolyai u. 13/a.)*
   ATARI-program kazettán
5. **Antal József** *(Sopron)*
   C64-program kazettán
6. **Vidács Jenő** *(Szarvas, Vajda P. u. 20.)*
   C64-program kazettán

A nyertesek ezúton is meghívást kapnak a KISZ KB KSZT 1987/88. tanévi szaktáborába.

A Középiskolai és Szakmunkástanuló Tanács Titkársága

Hangos billentyűzet

COMMODORE—64

Közismert, hogy a Commodore 64-en a billentyűk lenyomásakor nem kapunk hangvisszajelzést, és így könnyen elkerülheti figyelmünket, ha a lenyomott billentyűt a számítógép nem érzékeli. Ezen segít az alábbi gépi kódú program: a gong hangjára emlékeztető jelzés csak akkor következik be, ha a számítógép a lenyomott gomb hatására elfogadta az adatot.

Mint tudjuk, a C64 minden hatvanad másodpercben végrehajt egy belső megszakítást, melynek több célja van: ilyenkor kérdezi le a számítógép, hogy le van-e nyomva valamelyik billentyű, a program ellenőrzi a STOP billentyű lenyomását, villogtatja a kurzort, felfrissíti a belső órát stb. Ehhez a belső megszakításhoz csatlakoztathatunk, ha a $314—315 címen (788—789) található belső megszakítási vektor címét átírjuk úgy, hogy először a mi programunkra mutasson, majd a program végén visszaugrunk az eredeti belső megszakítási rutinra, amelynek belépési pontja $EA31.

Az *1. listán* a gépi kódú program látható. Maga a program a $C000 (=49152) fölötti memóriaterületen van, így a BASIC programok működését nem zavarja, és a NEW-utasítás is hatástalan rá. BASIC-ből a SYS(49152)-vel lehet elindítani és mindaddig működik, amíg a gépet ki nem kapcsoljuk, vagy a RUN/STOP és RESTORE billentyűket egyszerre le nem nyomjuk, vagy újra ki nem adunk egy SYS(49152) utasítást.

A listán bekeretezett számok megváltoztatásával egyrészt az 1. és 3. hanggenerátor frekvenciáját módosíthatjuk, másrészt a hang felfutásának és lecsengésének idejét. Ha például a középső bekeretezett szám — a $79 — helyett $7E-t írunk, akkor a lecsengés ideje 750 ms-ról 15 s-ra változik. POKE 49197, HANG1-gyel az első hanggenerátor frekvenciáját, POKE 49207, HANG3-mal a harmadik hanggenerátor frekvenciáját, POKE 49202, ATDEC utasítással pedig a hang felfutásának és lecsengésének idejét lehet módosítani. Így mindenki beállíthatja a számára legkellemesebb hangot.

A program természetesen BASIC-ből is a memóriába vihető (*2. lista*). A listán bekeretezett számokkal a hanggenerátorok paraméterei változtathatók meg: 28 az első hanggenerátor frekvenciáját, 121 a hang felfutását és lecsengését, 39 a harmadik hanggenerátor frekvenciáját szabja meg.

SZABÓ PÉTER PÁL

```
10PEN4,4
3 SYS(36864)
5 ;
7 ; GONG - HANGOS KLAVIATURA
9 ;
11 .OPT P4,00
13 ;
15 IDO = $05
17 KEY = $C5
19 IRQOLD = $EA31
21 IRQVEK = $314
23 SID = $D400
25 ;
27 *= $C000
29 ;
31 BE SEI
33 LDA IRQVEK
35 EOR #< SCAN ^ IRQOLD
37 STA IRQVEK
39 LDA IRQVEK+1
41 EOR #> SCAN ^ IRQOLD
43 STA IRQVEK+1
45 CLI
47 RTS
49 ;
51 SCAN LDA KEY
53 CMP #$40  ; VAN-E BILLENTYU LENYOMVA
55 BEQ NOKEY ; NINCS!
57 JSR SOUND ; HANGKELTES
59 LDA #IDO  ; AZ IDOZITES HOSSZA
61 STA CNTR
63 NOKEY DEC CNTR
65 BNE OUT   ; VISSZA A REGI IRQ-HOZ.
67 JSR NOSOUND; HANG KI
69 OUT JMP IRQOLD
71 ;
73 SOUND LDA #$1C ; FREKVENCIA
75 STA SID+1 ; HANG-1
77 LDA #$79  ; A HANG FELFUTASA 80MS,
79 STA SID+5 ; ES LECSENGESE 750MS.
81 ;
83 LDA #$27  ; FREKVENCIA
85 STA SID+15; HANG-3
87 ;
89 LDA #$0F  ; A HANGERO 15.
91 STA SID+24
93 ;
95 LDA #$15  ; HAROMSZOG HULLAMFORMA
97 STA SID+4 ; ES KORMODULACIO HANG 1-3
99 RTS       ;       KOZOTT.
101 ;
103 NOSOUND LDY #$1C
105 LDA #0
107 LP1 STA $D400,Y
109 DEY
111 BPL LP1
113 RTS
115 ;
117 CNTR .BYTE $05
119 ;
121 .END
```

1. lista

2. lista

```
1000 FOR I= 49152  TO  49233
1010 READ X:POKE I,X:S=S+X:NEXT
1020 DATA 120,173, 20,  3, 73, 34,141, 20
1030 DATA   3,173, 21,  3, 73, 42,141, 21
1040 DATA   3, 88, 96,165,197,201, 64,240
1050 DATA   8, 32, 44,192,169,  5,141, 81
1060 DATA 192,206, 81,192,208,  3, 32, 70
1070 DATA 192, 76, 49,234,169, 28,141,  1
1080 DATA 212,169,121,141,  5,212,169, 39
1090 DATA 141, 15,212,169, 15,141, 24,212
1100 DATA 169, 21,141,  4,212, 96,160, 28
1110 DATA 169,  0,153,  0,212,136, 16,250
1120 DATA  96,  5
1130 IF S<> 8526  THEN PRINT"ADATHIBA !":END
1140 PRINT"OK !!"
```

Lebegőpontos tömbök átadása BASIC→Pascal

Programjaink fejlődésében gyakran eljön az a pillanat, amikor az addig kényelmes BASIC túl lassú lesz, vagy más fogyatékosság miatt át akarunk térni Pascal nyelvre. Az alábbi program ahhoz nyújt segítséget, hogy ilyenkor legalább a numerikus tömböket ne kelljen újra adatokkal feltölteni, hanem egyszerűen át lehessen vinni az új programba.

Ehhez először is a két nyelv számábrázolási módját kell ismernünk.

A BASIC-ben a számokat 5 bájton tároljuk. Jelöljük ezeket abban a sorrendben, ahogy a memóriában vannak: B1, B2, B3, B4, B5-tel! Ebből a lebegőpontos szám kitevője a B1-en van tárolva eltolt formátumban, azaz 128 hozzá van adva:

expB = B1 − 128 (a BASIC kitevő)

A mantissza a B2 ... B5 bájtokon van, ahol a BASIC mantisszára igaz, hogy

1/2 ⇦ mB < 1

és a „kettedespont" a B1 7. bit előtt áll, amit mindig "1" követne, de rá van téve az előjel. Ezekkel a tárolt X szám:

X = előjel * mB * 2^expB

(speciálisan X = 0 esetén minden bájt 0.)

A Hisoft Pascalban ezzel szemben, ha a szám a P1, P2, P3, P4 bájtokon van ábrázolva, ahol P2 mutatja az exponenst, de a változatosság kedvéért kettes komplemens formában, azaz,

expP = P2, ha P2 < 128

expP = P2 − 256, ha P2 > = 128,

a mantissza pedig sorban a P4, P3, P1 bájtokon van, és a kettedespont P4 5. bitje előtt áll, a Pascal mantissza tartománya:

1 < = mP < 2.

Így a P4 6. bitje mindig 1, míg az előjel a BASIC-kel megegyezően a 7. biten áll. Az X szám tehát:

X = előjel * mP * 2^expP.

Mivel a mantisszának itt 1 és 2 között kell lennie, így az első átírási szabály:

mP = 2*mB, és emiatt

expP = expB − 1

Ha ez nem lenne, akkor a két exponens bájtot egyszerűen 128 hozzáadása kapcsolná össze (gondoljuk meg, hogy pozitív exp esetén a BASIC hozzáadott 128-at, ha megint adunk hozzá 128-at, akkor ugyanannyi lesz, mint eredetileg, és épp ez kell a Pascalban. Negatív exp esetén pedig a kétszer hozzáadott 128 éppen a 256-tal növelést adja), viszont mivel a két kitevő nem egyenlő, így a bájtokon végzendő művelet végeredményben:

P2 = B1 + 127

A mantisszát pedig jobbra kell léptetni 1 bittel, úgy, hogy B2-ben a 7. bit (előjel) marad, a 6. bitet pedig 1-be kell állítani. (X = 0 esetén pedig itt is minden bájt 0.)

```
 70 ;z/!!
 80 *L-
 90
100 ;lebegöpontos tömbök àtiràsa PASCAL formàra. 87.04.16.
110 ;magnò/dràjv
120 ;hivàsi formàja:
130 ;...USR ORG:REM vàltozònév
140 ;Elöször az adott vàltozò cimét megkeresi.
150
160 *D+
170 *L+
180         ORG  49801
190 ;hossza 147 byte
200 *L-
210
220         RST  32             ;a CHADD léptetése kétszer:
230         RST  32             ;":REM" àtugràsàval rààll a vàltoz
                                 önévre
240         CALL #28B2          ;a tömb cimét megkeresi
250         PUSH HL             ;és elteszi.
260         JR   NC,OK          ;ha megtalàlta, OK,
270         RST  8              ;egyébként Variable not found
280         DEFB 1
290
300 OK      RST  32             ;a CHADD-ot a sor végére tolja
310         CP   13
320         JR   NZ,OK
330
340
350
360
370 START   POP  HL             ;A tömb cime.
380         INC  HL             ;a név
390         INC  HL             ;és a teljes hossz àtugràsa
400         INC  HL
410         LD   B,(HL)         ;a dimenziòk szàma
420         INC  HL             ;elsö dimenziòméret cime
430         PUSH HL
440         LD   HL,4           ;(4 byte/szàm lesz)
450 ;a méreteket és 4-et összeszorzò ciklus
460 IDE     LD   (HOSSZ),HL     ;eddigi részletszorzat
470         POP  HL             ;i-edik dimenzió cime
480         LD   E,(HL)         ;mérete -> DE
490         INC  HL
500         LD   D,(HL)
510         INC  HL
520         PUSH HL             ;az i+1 -edik dimenzió cime
530         LD   HL,(HOSSZ)     ;eddigi részletszorzat
540         CALL #30A9          ;HL=HL*DE
550         DJNZ IDE            ;az összes dimenzió összszorzàsa.
560
570         LD   (HOSSZ),HL
580         PUSH HL             ;Pascal tömb hossza
590         LD   DE,50000       ;és kezdöcime
600         ADD  HL,DE
610         JR   NC,CONV0       ;ha elfér, mehet,
620         RST  8              ;egyébként Out of memory
630         DEFB 3
```

```
640
650 ;a tényleges átalakitó részlet
660
670 CONV0   POP   BC            ;Pascal tömb hossza
680         POP   HL            ;Basic tömb cime
690         PUSH  BC            ;visszatéréshez: Pascal hossz
700
710 CONV    PUSH  BC
720         CALL  #3297         ;egész számot valósra alakit
730         POP   BC
740
750 B1      LD    A,(HL)        ;a kitevő
760         CP    0             ;ha a kitevő 0,
770         JR    Z,ZERO        ;akkor minden byte 0!
780         INC   DE            ;P2 cime
790         ADD   A,127         ;B1 módositása
800         LD    (HL),A
810         LDI                 ;áttöltés P2-be
820         INC   DE
830
840 B2      SRA   (HL)          ;jobbra lép, de a 7. bit marad (el
                                 ő jel)
850         SET   6,(HL)        ;a 6. bit mindig 1
860         LDI                 ;áttöltés  P4-be
870         PUSH  DE            ;a köv. szám P1 cime lesz.
880         DEC   DE
890         DEC   DE            ;P3 cime
900
910 B3      RR    (HL)          ;jobbralép (átvitelt viszi)
920         LDI                 ;áttöltés  P3-ba
930         DEC   DE
940         DEC   DE
950         DEC   DE            ;P1 cime
960
970 B4      RR    (HL)          ;jobbralép
980         LDI                 ;áttöltés  P1-be
990         INC   HL            ;B5-öt kihagyjuk
1000        POP   DE            ;a következő számban P1 cime
1010
1020 NEXT   JP    PE,CONV       ;amig nincs kész (BC>0), tovább'
1030        POP   BC            ;kimenő adat: Pascal tömb hossza
1040        RET                 ;=======================
1050
1060 ZERO   LDI                 ;4 db 0 áttöltése
1070        LDI
1080        LDI
1090        LDI
1100        INC   HL            ;B5-öt kihagyjuk
1110        JR    NEXT          ;vége
1120
1130 HOSSZ  DEFS 2              ;PASCAL HOSSZ
1140
1150 ;++++++++++++++++++++++++++++++++++
1160
1170 ;drájv részére a PASCAL tömböt egy stringtömbbe tesszük
1180 ;hivási formája:
1190 ;...USR ORG1:REM változónév
1200 ;Először az adott változó cimét megkeresi.
1210
1220 ORG1   RST   32            ;a CHADD léptetése kétszer:
1230        RST   32            ;":REM" átugrásával rááll a változ
                                 ónévre
1240        CALL #28B2          ;a tömb cimét megkeresi
1250        PUSH HL             ;és elteszi.
1260        JR    NC,OK1        ;ha megtalálta, OK,
1270        RST   8             ;egyébként Variable not found
1280        DEFB 1
1290
1300 OK1    RST   32            ;a CHADD-ot a sor végére teija
1310        CP    13
1320        JR    NZ,OK1
1330
1340
```

Ezek alapján egyszerűen megírhatjuk a konvertáló programot. Először a gépi kódú részt nézzük *(1. lista)*.

A gépi kódú programot a 49801 címre tesszük, a keletkező Pascal tömb az 50000-től fog indulni (kb. 15 k nagyságú lehet maximálisan). Először egy ROM rutinnal megkerestetjük az adott változó címét, amit a HL-ben fogunk megkapni. Ez a részlet bármilyen változó keresésére más gépi programokba beépíthető.

A START című részben megnézzük, hány dimenziós a tömb, mekkora a kiterjedése az egyes dimenziókban. Így kiszámítjuk a Pascal tömb hosszát, ezt a HOSSZ címen is tároljuk, és a visszatérésnél a BC-be is betesszük, tehát ez lesz USR 49801 értéke.

A 710-es sornál kezdődik a tényleges átszámítás, az előzőleg megadott képletek szerint. Ezt a programrészt annyiszor hajtja végre, ahány eleme van a tömbnek. A HL regiszterpár mindig a kezelése alatt álló BASIC bájt címe, a DE a kapott Pascal bájtra mutat, míg BC visszafelé számolja az átalakított bájtokat; amikor 0 lesz, akkor van vége a munkának.

A 720-as sorban lévő utasítás fontos, ez ugyanis a BASIC tömbben lévő esetleges egész számot lebegőpontossá alakítja.

A B1 ... B4 címkék csak segítségképpen vannak bejelölve: mindig azt mutatják, hogy éppen a BASIC szám melyik bájtját alakítjuk át.

Az X=0 esetet a rutin a B1=0 alapján ismeri fel, ilyenkor a ZERO című részt hajtja végre.

A lista második részében (1220-as sor) a lemezmeghajtósok kedvéért van egy kis rutinocska, amely a Pascal tömböt egy BASIC karaktertömbbe másolja. A tömb egydimenziós, és azért tömb, nem pedig egyszerű sztring, hogy a BASIC-ben be lehessen előre állítani a pontos hosszát, ami, mint látni fogjuk, fontos.

Miután a sztringtömb címét megkerestük, az 1430-as sorban már a szöveg első karakterére mutatunk. Ide LDIR-rel áttöltjük a Pascal bájtokat (a HOSSZ-ból tudjuk, hogy mennyit).

A BASIC főprogram *(2. lista)* úgy használja ezeket a rutinokat — amelyek itt lebego-m2 néven vannak lefordítva —, hogy lemezmeghajtóval és magnóval is mehessen, ezért kétféle indítása van (lásd ehhez a 380-tól kezdődő sorokat is).

A BASIC tömböt olvastatjuk be ezután; a szokás szerint a lemezmeghajtó számát kérdezzük, és ha ez 0, akkor magnóról olvasunk.

A 160-as sor konvertál, és megtudjuk a Pascal tömb hosszát is. Ezután felvesszük a Pascal tömböt, az igény szerinti névvel. A névnek itt 8 karakteresnek kell lennie. Ha hosszabb, akkor visszadobjuk, ha rövidebb, akkor "-" jelekkel megtoldjuk. A Pascal szokásának megfelelően, a név 2. karakterében lévő ":" jelzi a meghajtót (lásd 220-as sor), és ekkor az első karakter a lemezmeghajtó száma. Jöhet tehát a felvétel akár magnóra, akár lemezmeghajtóra.

Ehhez a következőt kell tudni. A magnó esetén a Pascal megeszi a BASIC-ben simán Code-dal felvett blokkot, igy nincs sok problémánk (330—350-es sor). A le-

```
1350        POP   HL            ;A tömb címe.
1360        INC   HL            ;a név,
1370        INC   HL            ;a teljes hossz,
1380        INC   HL
1390        INC   HL            ;a dimenziók száma és az
1400        INC   HL            ;elsö dimenzióméret átugrása után
1410        INC   HL
1420 ;itt kezdödik a szöveg
1430        EX    DE,HL         ;a string lesz a cél
1440        LD    HL,50000      ;ez a forrás
1450        LD    BC,(HOSSZ)    ;Pascal tömb hossza
1460        LDIR                ;áttöltés
1470        RET
1480
1490 *L+
1500        END
```

1. lista

2. lista

```
  10 REM lebegopontos tomb konvertalas Basic->Pascal:_
 ------------------:
  20 REM --------start----------
  30 CLEAR 49800
  40 LOAD "lebego-m2"CODE
  50 GO TO 90
  60 CLEAR 49800
  70 LOAD *"m";1;"lebego-m2"CODE :
  80 REM ------Basic DATA ------
  90 INPUT "Basic file name ";b$
 100 INPUT "drive num. ";d: IF d=0 THEN  GO TO 140
 110 IF b$="" THEN  GO TO 90
 120 LOAD *"m";d;b$ DATA x()
 130 GO TO 160
 140 LOAD b$ DATA x():
 150 REM ----konvert--------
 160 LET hossz=USR 49801: REM x(:
 170 REM -------- SAVE -------
 180 INPUT "Pascal file name (d:..=drive) ";p$
 190 IF LEN p$>8 THEN  GO TO 180
 200 IF LEN p$<8 THEN  LET p$=p$+"_": GO TO 200
 210 REM drive OR tape?
 220 IF p$(2)<>":" THEN  GO TO 330
 230 LET d=CODE p$-48
 240 IF d<1 OR d>8 THEN  GO TO 180
 250 REM -------drive---------
 260 OPEN #4;"m";d;p$(3 TO )
 270 DIM x(1): DIM d$(hossz)
 280 RANDOMIZE USR 49915: REM d$
 290 PRINT #4;d$;
 300 CLOSE #4
 310 GO TO 90
 320 REM ---------tape----------
 330 SAVE p$CODE 50000,hossz
 340 BEEP 1,40: PRINT " RETURN  TO  VERIFY "
 350 VERIFY p$CODE 50000,hossz
 360 GO TO 90
 370
 380 REM ----szaporitas--------
 390 REM ---------tape----------
 400 SAVE "lebego.bas" LINE 30
 410 SAVE "lebego-m2"CODE 49801,147
 420 BEEP 1,40: PRINT " RETURN  TO  VERIFY "
 430 VERIFY "lebego.bas"
 440 VERIFY "lebego-m2"CODE
 445 STOP
 450 REM ------drive-----------
 460 SAVE *"m";1;"lebego.bas" LINE 60
 470 VERIFY *"m";1;"lebego.bas"
 480 SAVE *"m";1;"lebego-m2"CODE 49801,147
 490 VERIFY *"m";1;"lebego-m2"CODE
```

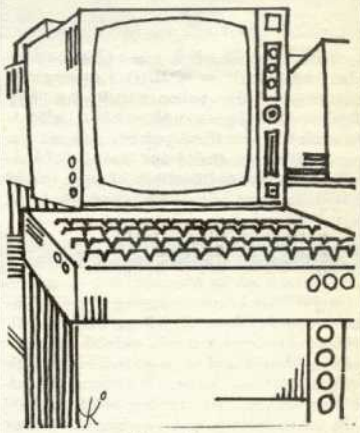
mezmeghajtón azonban a Pascal ún. szekvenciális fájlban tárolja az adatokat, tehát nekünk is ilyet kell produkálnunk. Ezért a 260-as sorban először nyitunk egy csatornát a kívánt nevű fájl számára. Az X tömböt lecsökkentjük, hogy legyen helyünk, majd a d sztringet pontosan a megfelelő hosszúra dimenzionáljuk. A 2. gépi kódú rutin ide teszi az átírt adattömböt, amit aztán a 290-es sorban a fájlba teszünk (a ";" nem hagyható el!).

Ne ijedjen meg senki, ha a gépi program működését nem érzékeli! Az lefutott, csak olyan gyorsan, hogy nem lehet észrevenni; máris a Pascal nevet kéri a program.

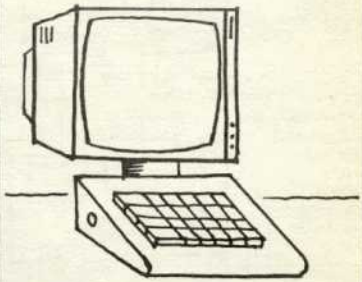
Mindkét variációban a felvétel után visszaugrik a program az elejére, újabb tömb beolvasására.

SLÍZ MIKLÓS

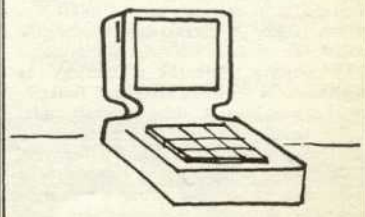
Hierarchia



főgebra



gebra



algebra

Miért nem stupid a MUPID? avagy: Milyen az intelligens videotex-dekóder? II.

Sorozatunk első részében röviden vázoltuk a MUPID koncepció lényegét, és felsoroltuk az így nyújtható, igen változatos alkalmazási lehetőségeket. Most a speciális vtx-dekóder belső felépítésével ismerkedünk meg, majd főbb üzemmódjainak ismertetését kezdjük el.

Felépítés és csatlakozó felületek

A MUPID központi egység egy 4 MHz-en ketyegő Z80A. 32 kB-os PROM-mal, 128 kB-os RAM-mal (vtx-oldalak és programok tárolására) és a vezérlőkarakterek számára külön 8 kbájtos RAM-mal rendelkezik *(1. ábra)*. Készüléken belüli további 128 kB-os bővítésre is van lehetőség. Egy másik mikropocesszor figyeli és alakítja a billentyűzetről érkező jeleket soros adatjelekké a dekóder számára, és ugyancsak erre van bízva a billentyűzet kódolása, a pergésmentesítés és tartós kiváltás, valamint a hangkeltés. A MUPID A1 és A2 jelű analóg bemeneteit (botkormányt is) fogadó A/D átalakító kezelését is ez a mikroprocesszor végzi. A billentyűzet fóliaérintős megoldású, az alfanumerikus billentyűktől különválasztva találhatók a számjegy- és hagyományos videotex lekérdező funkciókat (központ hívását, oldal lekérését, oldal ismételt lekérését, felfedését, attributum hatástalanítását stb.) megvalósító billentyűk. A készülékbe tápegységet és hangszórót is beépítettek. Utóbbi elsősorban a billentyűlenyomás akusztikus visszajelzésére szolgál, de zenélő programok is megszólaltathatók rajta — természetesen gyenge minőségben. A központi egység egy SIO áramkörön keresztül tartja a kapcsolatot a külső perifériákkal: a SIO programozását a felhasználó számára is lehetővé tették. A MUPID-ot a hozzá csatlakoztatható kiegészítő berendezések teszik még sokoldalúbbá. A tervezők olyan interfészeket alakítottak ki a dekóderen, melyek sokféle külső eszköz használatára nyújtanak lehetőséget. Lássuk tehát a csatlakozó felületek *(2. ábra)* főbb jellemzőit.

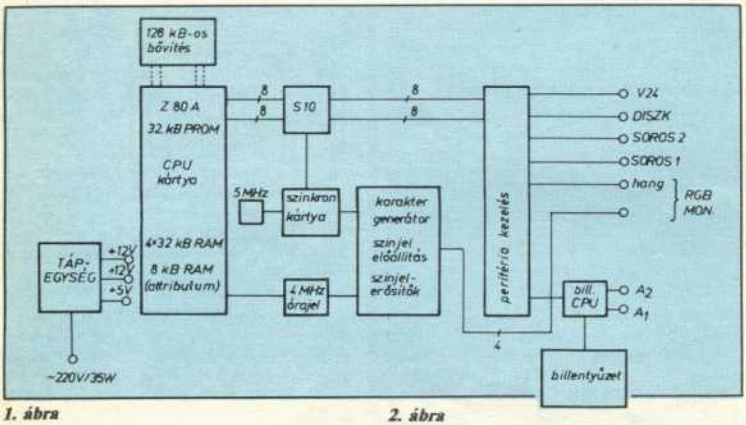
A **TV/RGB monitor** csatlakozás a széles körben terjedő EUROSCART/CENELEC csatlakozó, mely az R, G, B alapszín-összetevőn kívül a kioltójelet, összetett szinkronjelet és a hangjelet szolgáltatja.

Az **A1, A2 analóg** bemenetek 6 pólusú DIN aljzatok, elsősorban botkormányok csatlakoztatására vezették ki őket, de alkalmasak 0— +5 V feszültségtartományba átalakított, tetszőleges analóg mennyiségek fogadására is.

Az **1. soros** interfész 9 pólusú szubminiatűr ISO aljzat, ide csatlakozik a speciális vtx-modem. A felületen kereszül soros aszinkron adatátvitel 7 bites (1 start-, 7 karakter-, 1 stopbit) vagy 8 bites (1 start, 8 karakter-, 1 stopbit) adatformátumú jelekkel valósítható meg, alapállapotban 75 bit/s adási és 1200 bit/s vételi sebességekkel. Szimmetrikus 300, 600, 1200, 2400 és 4800 bps adatátviteli sebességek is programozhatók. A csatlakozóra természetesen a modemet vezérlő jeleket is kivetették, az elektromos jelszintek nem szabványosak, a modulációs jellemzők V. 23. ajánlás szerintiek.

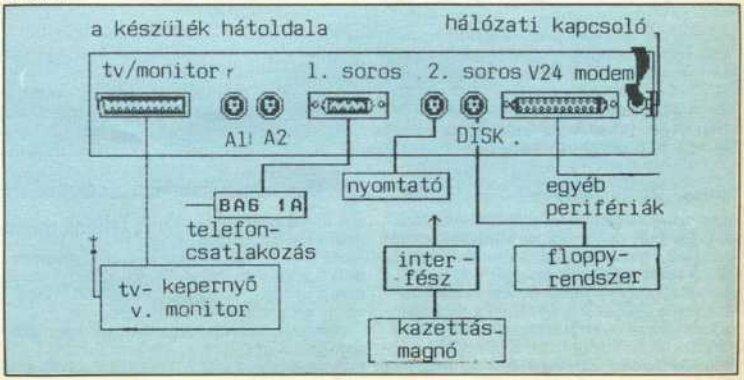
A **2. soros** interfész 8 pólusú DIN aljzat, aszinkron, 1200, 2400 vagy 4800 bps adatátviteli sebességre programozható. A jelszintek V. 28. ajánlás szerintiek, az adatformátum 8+2 bit. Közvetlenül ide csatlakoztatható a MUPID grafikus tábla vagy többféle, soros bemenetű nyomtató, vagy a MUPID cég által gyártott külső adapteren keresztül tetszőleges gyártmányú kazettás magnó.

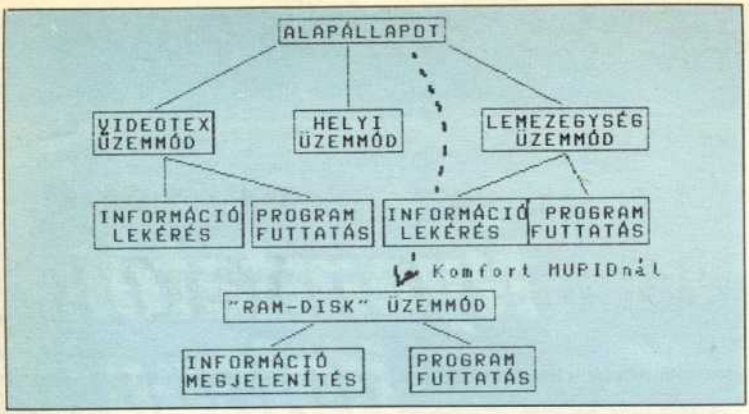
A **"DISK"** feliratú csalakozó 8 pólusú DIN aljzat. Kivezetései és azok funkciói hasonlóak a 2. soros interfészhez. Elsősorban a szintén MUPID-gyártmányú M-DISK mágneslemezegység fogadására szolgál.



1. ábra

2. ábra





3. ábra

A **V. 24-es interfész** 25 pólusú, szabványos lábkiosztású. A jelszintek a V.28. ajánlásnak megfelelőek. A 9-es lábon a belső +5 V is megjelenik, 100 mA terhelhetőséggel. Az adatformátum itt is 8+2 bites, az átviteli sebesség 300—9600 bit/s között szoftverből állítható, az 1200/75, 1200, 2400 és 4800 bit/s sebességek a billentyűzetről menüből is kiválaszthatók.

Az osztrák MUPID háromféle változatban kerül forgalomba:

— Standard-MUPID — a belső felépítés a fent leírtaknak felel meg
— Komfort-MUPID — beépített, akkumulátorral pufferelt 128 kB-os CMOS RAM bővítéssel
— Digi-MUPID — beépített digitalizáló modullal

Ez utóbbira a digitalizáló konfiguráció ismertetésénél még kitérünk. A standard változatból utólag bármelyik más változat létrehozható, de memóriabővítővel és digitalizáló modullal együtt nem építhető be. A továbbiak mindhárom változatra érvényesek, a komfortkiépítést igénylő alkalmazásokat külön említjük meg.

A MUPID üzemmódjai

A MUPID-ot elsősorban a videotex alkalmazások szemszögéből tágyaljuk, ezen belül is az osztrák vtx-rendszerben való viselkedését. A dekóder lehetséges üzemmódjait mutatja a *3. ábra*. Ezeket vagy a felhasználó a billentyűzetről, vagy valamely futó program automatikusan választja ki. A szükséges elemek: MUPID dekóder bármelyik változata, monitor, vtx-modem és M-DISK mágneslemezegység.

Videotex üzemmód

Az 1. soros interfészhez speciális vtx-modemet — azt pedig meglévő telefonkészülékekkel sorosan a távbeszélő-hálózathoz — csatlakoztatva, a SHIFT és a vtx-központot hívó billentyűk együttes lenyomását követően a modem automatikusan felhívja hívószámát vagyis a vtx-központot, amelyet a PROM-jában tárol. A MUPID alapértelmezésben a vtx-modem automatikus híváskezdeményezésére van programozva, s a központtal való kapcsolat felépülte után 75 bit/s sebességgel adja — és 1200 bit/s-mal fogadja — az adatokat. A billentyűzetről megfelelő paranccsal a képernyő 25. sorában megfelelő menü hívható be, amely lehetővé teszi ugyanezen modemmel a manuális hívást, vagy a V. 24-es modem használatával különböző (1200/75, 1200 DX, 2400 DX és 4800 DX) adatátviteli sebességek közötti választást. Ilyenkor a felhasználó dolga, hogy tárcsázzon, és ha a vtx-központtól megjött a megfelelő válaszhang, a billentyűzetről a vonalra kapcsolja a modemet. A sikertelen hívásról — legtöbbször annak okát közlő — üzenet jelenik meg a 25. sorban.

Sikeres hívásra a hívószámhoz tartozó vtx-központ udvariasan bejelentkezik, majd a billentyűzetről bevitt megfelelő karakterekkel azonosíthatjuk magunkat a rendszer számára. Ezután lehetséges a hozzáférés a hőn óhajtott információt tartalmazó oldalhoz (menüből való választások útján egyre közelebb juthatunk a célhoz), vagy a keresett oldal számát — ha ezt ismerjük — csillag és kettőskereszt jelek közé zárva közvetlenül megadhatjuk a központ számára. Utóbbi esetben az oldalszám a 24. sorban megjelenik, de ezt nem a dekóderünk küldi a monitorra, hanem az előbb eljut a vtx-központig a 75 bit/s-os csatornán, majd onnan 1200 bit/s sebességgel, mintegy „visszatükrözve" érkezik vissza a dekóderbe, majd onnan a képernyőre. Ily módon ellenőrizhető, hogy helyesen billentyűztük-e be a gyakran 8-10 karakter hosszúságú számot, és hogy azt a központ megfelelően vette-e. Törölni a csillagjel ismételt beadásával lehet. Tehát így hívhatjuk le a megjelenítésre szánt információs oldalt, de az információszolgáltatók „válasz"-oldalát vagy más előfizetőnek szánt elektronikus levél (Mail-Box) oldalát is.

Teleprogram lehívásánál a kék K (Kommandó-parancs) billentyűt majd az A betűt kell előzőleg lenyomnunk, mielőtt a teleprogramot tartalmazó oldal első aloldalát közvetlenül vagy menüből lehívnánk. Így szerez a MUPID tudomást arról, hogy nemcsak a megjelenítendő oldal adatjelét kell fogadni, hanem teleprogramot és azt más memóriaterületre kell betöltenie. (Általában a teleprogram betöltése alatt — melynek hossza jelenleg legfeljebb 24 kB lehet — a felhasználó szives türelmét kérő és a betöltés várható idejét közlő felirat látható.) Az így betöltött teleprogram vagy automatikusan elindul, vagy a képernyőn látható utasítással — általában egy billentyű lenyomásával — indítható. A teleprogramok helyileg futtathatók, így az összeköttetés a vtx-központtal megszakítható. Vannak olyan programok is, melyek futásuk közben a vtx-központhoz fordulnak adatokért, így ezeknél a telefonvonalat nem szabad bontani. Az osztrák vtx-központban a MUPID cég több, a MUPID dekóder és a központ együttműködését, valamint a felhasználó komfortosabb kiszolgálását segítő rendszerprogramot helyezett el, melyek egy része a kék K és valamely más billentyű lenyomását követően azonnal, automatikusan betöltődik a dekóderbe. Ilyenek például:

— K I olyan „kulcsszavas" keresőprogramot hív le, amely kikeresi azt az oldalt, ahol egy ABC szerint rendezett listán a keresett téma szerepel és onnan közvetlenül lehívható,
— K M későbbi hívás könnyítése céljából a kért oldalak 0—9-ig megjelölhetők,
— K E szerkesztőprogram lehívása
— K B BASIC program lehívása.

Más rendszerprogramok — nyomtató, másoló, levélszerkesztő stb. — a többi teleprogramhoz hasonló módon tölthetők be.

A központot hívó billentyűt önmagában lenyomva, a MUPID és a központ közötti kapcsolat bármikor megszakítható. Ha a modem lebonyolította az összeköttetést, a sorba kötött telefonkészülék ismét alkalmas a hagyományos telefonbeszélgetésre, a MUPID pedig visszatér alapállapotába.

(Folytatjuk)

JURENKA OSZKÁR

Nyomtatók holnapja...

A szép iráskép mellett a viszonylag kis zaj is jellemzőjük a mátrixprintereknek. Így alkalmasak irodában, gép melletti üzemeltetésre. Manapság szinte csak a szocialista országokban gyártanak fix karakterkészlettel rendelkező berendezéseket. A leggyakoribb típusok „letölthető" karakterkészlettel rendelkeznek. Sőt, a programozási lehetőségeket kihasználó szoftverekkel (ezekből nálunk a LETRIX és a FONTRIX a közismert) lehetőség van változatos betűtípusú, akár magyar ékezetes szövegek előállítására is. A FUJITSU 2400-as típusú nyomtatója a nyugat-európai piacon nagyfokú programozhatóságával tűnik ki. Ezt megkönnyíti a szinte luxusszámba menő 16 karakteres saját LCD displaye.

A nyugat-európai piacon egyre inkább csak tartozékokkal — például lapadagolóval — ellátott nyomtatókat lehet professzionális célra eladni. Így előállt ugyanaz a helyzet, amely korábban a PC-piacot jellemezte. Az alapgép viszonylag olcsó, de a tartozékok ára összemérhető a berendezésével. Ennek ellenére az ilyen tartozékok alkalmazása a gyakorlatban kifizetődik, mert a hagyományos leporellóknál olcsóbb irodai papírt, illetve hagyományos átírós űralapokat lehet használni.

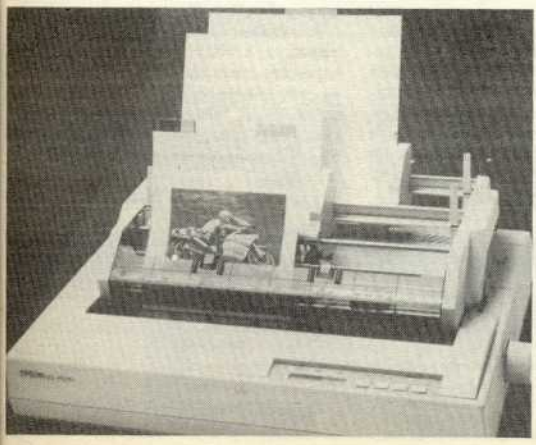
A lapadagolók gyártói szinte külön „kasztot" alkotnak. Igen jó hírnévre tett közülük szert a RUTISHAUSER, valamint az írógépgyártásáról ismert HERMES-cég.

Az elektronika szinte a végtelenségig továbbfejleszthető. S ez a fejlesztés az egyre terjedő és viszonylag olcsó berendezésorientált mikrochipek segítségével gyorsan és elfogadható költséggel végrehajtható. Ennek szembeszökő példája az OKIDATA 393-as nyomtatója, amely cserélhető emulációs (IBM, EPSON, QUME stb.) modulokkal rendelkezik.

Teljesen új elven alapulnak az új, igen nagy sebességű mátrixnyomtatók. Közülük is kiemelkedik a 800-900 sor/perc (mintegy 1970 karakter/másodperc!) sebességű SHUTTLE típusú berendezés. A teljes nyomtatási mezővel szemben egy fésűfogakból álló nyomtatófej helyezkedik el, egy fog/egy karakter sűrűséggel. A fog végén lévő kidomborodás pedig attól függően hagy nyomot a papíron, hogy a hozzá tartozó tekercsben folyik-e áram. Mivel karakterenként egy fésűfog van, a nyomtatófejet vízszintes irányban csak egykarakternyit kell elmozdítani. Így sikerült egyesíteni a mátrixnyomtatók rugalmas karakterkészletét valamint a nagy sebességű nyomtatást, ami eddig csak a karakterszalagos-kalapácsos professzionális nyomtatók sajátja volt, de ott is fix karakterkészletet lehetett csak használni. Ugyanezt az elvet alkalmazza a GENICOM 4410-es és 4440-es típusú valamint a PRINTRONIX P 600 XQ jelű nyomtatója. Ezek a nyomtatók a képet ütéssel alakítják ki, s így 4-5 példány kifogástalan átírására alkalmasak önindigós papíron.

Új elven alapulnak a „resistive ribbon" elvű printerek. Ilyen például a HERMES cég 810-es típusjelet viselő szerkezete. Ennél a típusnál a festékszalag elektromosan vezető, viaszba ágyazottt festéket tartalmaz. Az írófej egy 40 elektródát tartalmazó függőleges tűsor, ahol az egyes elektródák (képpontok) átmérője 0,05 mm. Az elektromosan vezető szalagon és az elektródán keresztül áram folyik, amely helyileg felmelegíti a szalagot, és a festék egy 0,1 mm átmérőjű foltban a papírra olvad. Visszamenve

Az EPSON SQ—2500-as típusú tintasugaras nyomtatója e kategóriában megfelelője az LQ—2500-as hagyományos tűmátrixos nyomtatónak. Programozási sajátosságai — így még betölthető karaktergenerátora illetve az LCD kijelzője is — azonosak azzal. Újdonsága, hogy 540 jelet tud kinyomtatni karaktergenerátoros üzemmódban másodpercenként. Ami megkülönbözteti, hogy tintasugaras nyomtatóműve révén, újságminőségben képeket, grafikákat lehet vele előállítani 109 pont/milliméteres felbontással. A nyomdai minőségű betűk generálásánál viszont kissé lassú, csak 180 jelet állít elő másodpercenként.



Az EPSON GQ—3500-as típusú lézernyomtatója, klasszikus konstrukció a maga kategóriájában. A félvezető henger, az előhívó, valamint a festék egyetlen kazettában egyszerre könnyen cserélhető. A kis teljesítményű (irodai) berendezések közé tartozik, percenként 6 oldal teljesítményével. Az alapfunkciókat az oldallapon lévő kezelőszerveivel lehet beállítani, de természetesen programból is vezérelhető. Egyaránt alkalmas a beépített karaktergenerátorral valamint a RAM-ba betöltött karakterkészletekkel való nyomtatásra, illetve grafikus üzemmódban való üzemeltetésre. Színes nyomatot a tónert, előhívót és a hengert tartalmazó kazetta egy kézmozdulattal történő cseréjével állíthatunk elő — természetesen csak egy színben. Jelenleg kék, zöld, piros, barna valamint fekete festékkazettát forgalmaznak. A nyomtató emulációs üzemmódja (HP, IBM, EPSON stb.) az IC kártyára dugaszolható ROM segítségével történik.

Ha körültekintünk valamelyik nagy számítástechnikai kiállításon, észrevehetjük, hogy a hagyományos 9 vagy 18 tűs nyomtatók piaca telítetté vált. A most forgalomban lévő típusok közös jellemzője a szép írásképe, a megbízható elektronika, a viszonylag egyszerű programozhatóság. Bizonyos szabványok is kialakultak. Így a legtöbb gyártó nyomtatóit IBM vagy Epson kompatibilisre készíti. Ennek megvan az az előnye is, hogy a programcsomagok gyártói ehhez tudnak alkalmazkodni. Ez újabb piaci keresletet jelent.

a kinyomtatott szövegre s alacsonyabb hőmérsékleten futtatva az írófejet, a festék nyomtalanul visszaszedhető a festékszalagra. Nyomtatási sebessége mintegy 40 karakter/s. Csak a fejmozgató motor zaja jelentkezik, ezért igen csendes üzemű. Hátránya, hogy az írás fényes felületű papírról viszonylag könnyen lekopik. Ugyanakkor kifogástalanul fotózható írásképet nyújt.

Hasonló átviteli elvet alkalmaznak, csak a festékszalag többszínű, az úgynevezett „thermal transfer" átviteli eljáráson alapuló nyomtatók. Mintegy 1 perc alatt kialakul a 3 alapszínt és néhány esetben a feketét is tartalmazó viaszos festékszalagok és a hőíró fej segítségével a mintegy 4000 színárnyalatot tartalmazó színes kép. Gyakorlatilag az utolsó előtti lépés a fotóminőségű színes másolatok kialakításához! Jelenleg üzemeltetésük igen drága. RGB interfésszel rendelkeznek, így a színes grafikus kártyák jelével vezérelhetők. Jelenleg főként a MITSUBISHI G 650 és a MITSUBISHI G 500A nyomtatók uralják a — nem éppen nagy — piacot.

Érdekesség, hogy az ismert „tintaágyús" nyomtatási elvet hasznosító, úgynevezett „Jet ink" nyomtatók is tudtak színes újdonsággal szolgálni. A jelenlegi „sláger" a DATAPRODUCTS SI 480 SOLID INK nyomtató. Benne a megszokottól eltérően a tinta szilárd pasztillában kerül betöltésre, és csak a nyomtatófej hőjének hatására válik folyékonnyá. 32 függőlegesen elhelyezett fúvókán keresztül csapódik a papírra a tinta. Jelenleg csak a fekete változatát gyártják, a színes most van fejlesztés alatt. 480 pont/inch a felbontása, sebessége 400 karakter másodpercenként.

Egyre többen gyártanak xerografikus elven működő nyomtatót (lézerprintert). Itt a mindennapi gyakorlat az, hogy néhány nagy gyártó nyomtatóművét építik össze saját fejlesztésű elektronikával. A legeltérőbben beépített nyomtatóművet a TOSHIBA készíti (fő felhasználói a Dataproducts és a WEGNER), míg mások általában Canon (Mannesmann Tallya) vagy éppen Xerox nyomtatóművet alkalmaznak. A képet lézer forgótükörrel vagy piezokristállyal vezérelt rezgőtükörrel eltérített sugara alakítja ki a szerves félvezető vagy szelénhenger felületén. Sebességük általában 8-40 lap/perc teljesítményig terjed.

Újdonságszámba megy a kialakult konstrukciós elvek között a CENTRONICS PAGE PRINTER 8 típusú nyomtatónál, hogy a hagyományos szelénhenger helyett a félvezető réteget egy fóliára vitték fel. Technológiai újdonságot alkalmaz a LIQUID CRYSTAL SHUTTER nyomtató — amelyet a hannoveri CeBIT kiállításon láthatott először a közönség: egy folyadékkristályos fényrekesz és egy hagyományos megvilágító pontlámpa kombinációjával alakítja ki a képet a fotofélvezetőt tartalmazó henger felületén. Nem alkalmaz lézert. Felbontása ennek ellenére igen jó, 300 pont/inch.

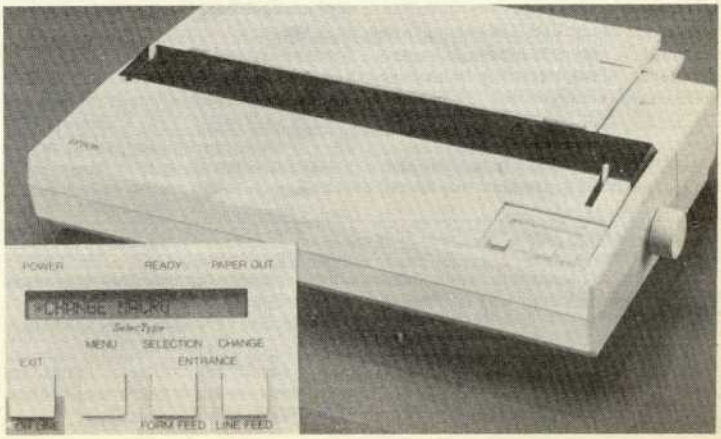
Érdekesség a lézernyomtatók működési elvét felhasználó, de nem lézeres nyomtatómű, a C.ITOH CIE 3000-es. Itt a lézer helyét egy ionokat kibocsátó fej veszi át, amely létrehozza a félvezető hengeren a töltésképet. A továbbiakban minden a lézernyomtatóknál megszokott xerox elven történik.

Az abszolút csúcs a nyomtatók piacán jelenleg a japán TOSHIBA cég színes digitális nyomtató-másoló berendezése, amelynek nyomtatóműve valószínűleg sok készülékben fog megjelenni az elkövetkező években.

A karakterszintetizáló nyomtatók — a grafikus lehetőség, a könnyen programozható, cserélhető karakterkészlet miatt — nagy jövő előtt állnak. Ugyanakkor a fix karakterkészlettel rendelkező — karakterszalagos, gömbfejes, margarétafejes, karakterdobos — nyomtatók, bár belőlük készül jelenleg a legtöbb, rövidesen a számítástechnika őskövületei közé fognak tartozni, s rövidesen eltűnnek a gyártók, kereskedők termékpalettájáról. *(VIDEOTON RT. INFO nyomán)*

KIS JÁNOS

Az LQ—2500 24 tűs nyomtató, egyike az EPSON legkorszerűbb berendezéseinek. Normál — karaktergenerátoros — üzemmódban 324 karaktert ír másodpercenként, betölthető karaktergenerátora egyszerre öt betűkészletet képes alkalmazni. A programozást saját LCD display könnyíti meg. Levélminőségű nyomatot ad. Soros (RS 232C) valamint párhuzamos (CENTRONICS) interfésszel rendelkezik.



Szoftver

Cikksorozatunkban először a gépi kódú programozást segítő három legfontosabb eszközzel, az *EDITOR*, az ASSEMBLER és a MONITOR programmal kapcsolatos alapvető ismereteket mutatjuk be. Aki gépi nyelven akar programozni, annak ezekre a programokra feltétlenül szüksége van.

Az EDITOR

A TEXT EDITOR-okról általánosságban

A TEXT EDITOR magyarul szövegszerkesztőt jelent. Segítségével a felhasználó szövegfájlokat hozhat létre, illetve a már meglévőket módosíthatja.

Szövegfájloknak nevezzük azokat az adathalmazokat, amelyek karaktereket tartalmaznak, és megjeleníthetők a számítógép képernyőjén vagy nyomtatón kinyomtathatók. Szövegfájl például egy Z80 assembly nyelven megírt ún. forrásnyelvi program, egy programhoz tartozó, az adathordozón (szalagon vagy lemezen) lévő leírás vagy kezelési útmutató.

Az EDITOR a számítástechnikában az ember-gép kapcsolat megvalósításának legfontosabb eszköze. Az iratok, levelek, dokumentációk számítógépes kezelését biztosító irodai szövegszerkesztő rendszereknek is alapvető, legfontosabb programja. Mivel használata programozási ismereteket nem igényel, kitűnő lehetőséget teremt a számítástechnikával való megismerkedésre.

Hogyan működik az EDITOR program?

Kezdetben az ún. parancs módban arra vár, hogy a felhasználó megadjon egy parancsot. Miután ez megtörtént, átvált a végrehajtási módba, végrehajtja a parancsot, majd ismét parancs módba vált, és ez a folyamat ismétlődik.

A felhasználó az EDITOR-nak szóló parancsokat és a bevinni kívánt szöveget a számítógép billentyűzetén írja be. A begépelt karakterek megjelennek a képernyőn. Ezt a visszairási módot echozásnak (echo = visszhang) nevezik.

Az EDITOR a szöveges információ alábbi egységeivel tud dolgozni:

— betű (karakter),

— szó (elején és végén betűközzel elválasztott karaktersorozat),

— mondat (a mondathatároló jeltől (. ! ?) a következő ilyen jelig tartó karaktersorozat),

— bekezdés (a RET karaktertől a következő RET karakterig tartó karaktersorozat),

— blokk (egy kijelölt szövegrész).

A megadható parancsokat hat csoportra oszthatjuk:

— munkaterület-kezelő (buffer management)

— mutatómozgató (pointer move)

— szövegbeillesztő (insert)

— szövegkereső (search)

— szövegtörlő (delete)

— egyéb speciális parancsok

Munkaterület-kezelő parancsok

Az EDITOR az ún. munkaterületen végzi a szövegszerkesztést. A munkaterület az írható-olvasható memória egy darabja, ide kell elhelyezni a módosítandó vagy beírandó szövegfájlt. A munkaterület mérete véges, így gyakran megesik, hogy a teljes szövegfájl nem fér el benne. Ezért a nagyméretű fájlok szerkesztéséhez az ún. lapozási technikát alkalmazzák.

Ennek az a lényege, hogy a munkaterületre betöltik a szövegfájl akkora darabját, amekkora éppen elfér ott, majd az ezen a részen végzendő javítások végrehajtása után a módosított részt elmentik, és a munkaterületre a fájl következő részét töltik be. Egy lapnak nevezik a szövegfájl munkaterületre betöltött részét. Ezzel a megoldással elméletileg végtelen méretű fájlok szerkesztése is megoldható.

A szövegfájl soron következő lapját a forrásbemenetről tölthetjük be a munkaterületre. A forrásbemenet lehet kazettás magnó vagy hajlékonylemez. A megszerkesztett vagy megírt lapot a munkaterületről a forráskimenetre menthetjük, ami az előbbi perifériák valamelyike lehet.

A munkaterületet kezelő tipikus parancsok a következők:

— a munkaterület feltöltése a forrásbemenetről,

— a munkaterület egészének vagy részeinek kimentése a forráskimenetre,

— lapcsere (a munkaterület kimentése, majd a soron következő lap betöltése),

— a munkaterület listázása (ennek során a munkaterület tartalma megjelenik a képernyőn vagy a nyomtatón).

Mutatómozgató parancsok

Mivel a munkaterületen a fájl egy tekintélyes darabját lehet egy időben tárolni, az EDITOR számára itt meg kell jelölni azt a pontot, amelytől kezdve végrehajtsa a megadott parancsot. Az erre szolgáló eszközt mutatónak (pointer) nevezzük. Ez tulajdonképpen egy, a tárolóban őrzött számérték, amelynek értékkészletét a munkaterületen tárolt karakterek lehetséges címei adják.

Ha az EDITOR mutatóját betűnként (karakterenként) lehet mozgatni, akkor karakterorientált, ha csak soronként lehet mozgatni, akkor sororientált EDITOR-ról beszélünk. A tipikus mutatómozgató parancsok a következők:

— a mutató megadott számú karakterrel/sorral előre mozgatása,

— a mutató megadott számú karakterrel/sorral vissza mozgatása,

— a mutató beállítása a megadott című karakterre/sorra,

— a mutató beállítása a munkaterület elejére (a legelső karakter/sor elé),

— a mutató beállítása a munkaterület végére (a legutolsó karakter/sor mögé).

A munkaterület végén általában nem a puffer fizikai végét, hanem a karakterekkel még fel nem töltött munkaterületrész elejét értjük.

Szövegbeillesztő parancsok

Szövegbeillesztésnek (insert) nevezzük azt a műveletet, amelynek során beillesztjük a munkaterületre a begépelt karaktereket. Az új karakterek az aktuális karakter (sor) elé kerülnek, amely az utána következőkkel együtt a munkaterület vége felé elmozdul.

Szövegtörlő parancsok

A szövegtörlés (delete) során a felhasználó által megadott számú karaktert (sort) kitöröljük a munkaterületről. A kitörölt részt követő karakterek a munkaterület eleje felé elmozdulnak.

Szövegkereső parancsok

Szövegfájlok javításánál gyakran nem tudjuk pontosan, hogy a javítandó sor vagy karakter a fájl elejétől számított hányadik sorban helyezkedik el. A javítás helyének környékén azonban többnyire található valamilyen jellegzetes szövegkörnyezet. Ilyen esetekben jól használhatjuk a szövegkereső parancsok valamelyikét. Miután megadtuk a keresett szöveget — általában maximum 16—32 karakter lehet —, az EDITOR a munkaterületen megkeresi azt. Ha megtalálja, rendszertől függően vagy a keresett szöveget tartalmazó sor elejére (sororientált) vagy a keresett szöveg mögé (karakterorientált) állítja a mutatót. Ha a keresés sikertelen — az adott szöveg nem fordul elő a munkaterületen —, a rendszer jelzi ezt.

A szövegkereső parancsok csoportjába sorolhatók a szövegcserélő (text-change) parancsok is. Hatásukra az EDITOR megkeresi a megadott régi szöveget, törli a munkaterületről, és helyére illeszti az ugyancsak begépelt új szövegrészt.

Speciális EDITOR parancsok

A különféle EDITOR-ok az általános parancsokon túlmenően több-kevesebb speciális parancs végrehajtására is képesek. Ezek célját és felhasználási módját a felhasználási leírásokból ismerhetjük meg.

Megjegyzés

A fentiekben általános képet mutattunk az EDITOR-okról. A jelenleg használt korszerű szövegszerkesztők már kizárólag karakterorientáltak, egy kijelzett képernyőnyi szöveg tetszőleges része szerkeszthető, a kurzorral a képernyő teljes felületén mozoghatunk. Ennek az az előnye, hogy a szövegen azonnal látható a módosítás. Ezek a szövegszerkesztők speciális alkalmazásokra „kihegyezhetők", azaz hozzáigazíthatók egy-egy programnyelv szövegének beviteléhez, ami azt jelenti, hogy egy-egy speciális billentyű megnyomásakor egy egész — például nyelvi — alapszó jelenik meg a szövegben. Az egyre nagyobb méretű belső memóriák nagy munkaterületek használatát teszik lehetővé.

TISZAI TAMÁS—
DR. KÓNYA LÁSZLÓ

Ígéretes kezdet

Volt egyszer egy kisfiú, egy nyár és egy számítógép. Az első és az utolsó a középsőben talált egymásra. Hogyan történt az ismerkedés? Tartózkodással indult, aztán a kíváncsiság, a rajongás következett. Minden idejét, figyelmét lekötötte a számítógép nyújtotta élmény.

A profi számítógépesnek mindez nem sokat mond, hiszen számára természetes és nem csoda a programkészítés, a programnyelv használata. De kérdem én, egykoron nem mindenki hasonlóan kezdte? Senkinek sem hullott ölébe a mostani tudás. Játék, igenis csoda, csodajáték lehetett a számítógép, az, ami Gönczi Dávidnak, a nyolcesztendős kisfiúnak ma.

— Anyu kérdezte a nyáron, akarok-e számítógépes táborba menni. Olyanba, ahol sokat játszhatok, meg még úszhatok is. Osztálytársaimtól hallottam, mennyire klassz dolog a számítógépezés, hogy fél napokat elszórakoznak a különböző játékprogramokkal. Egyszer egy barátom hívott, hogy náluk játsszunk, neki van ugyanis Commodore 64-e. Aztán elmaradt az egész, mert beteg lettem. De azért nagyon kíváncsian vártam a találkozás pótlását.

Az Almássy Téri Szabadidőközpontban a Compania társaság szervezte nekünk, gyerekeknek ezt a vakációs programot. Egy időben húsznál is többen gyűltünk össze a számítógépes teremben. Az alsósokkal és a felsősökkel külön foglalkoztak, úgy, hogy más és más feladatokat adtak nekünk. Egyetemista bácsik vigyáztak ránk, és szintén ők magyarázták el a gépek használatát.

Választék bőven volt, kinek melyik tetszett, ahhoz ült. Én mindjárt a C Plus/4-et választottam, bár akkor még nem is tudtam, hogy az milyen. Egy ismerősöm a HT-t nézte ki, egy másik a Videoton TV-Computert. De ezeken kívül volt ott még C64, C16 és Primo is. Nem tudom miért a Plus/4 mellett döntöttem, hirtelen született az elhatározás. Egy-egy géphez két gyereket osztottak, s ez azért is jó volt, mert miután elmagyarázták a gép tulajdonságait, a billentyűk jelentéseit és kezelését, akkor kezdődhetett a játék. A párommal csudára drukkoltunk egymásnak. Borzasztóan izgultunk, hogy mi nyerjünk, méghozzá nagyon sokat, s ne a számítógép fogjon ki rajtunk.

Azután amikor már jól kiszórakoztuk magunkat, akkor az egyetemista bácsi megtanított bennünket a programozás alapjaira. Naponta másfél óránk telt el így, vagyis ez az idő rendkívül hasznos volt. Sikerült olyan játékosan tanulnunk, hogy észre sem vettük azt, hogy ott oktatás folyt. Mindenesetre az egészet nem úgy fogtuk fel, mint a szigorú iskolai tanulást. Talán ezért haladtunk gyorsabban és eredményesebben.

Nem tagadom, a legjobb mégiscsak a játék. Viktor barátommal eleinte a Timeslip játékprogram volt a kedvencünk. Eleinte persze követtünk el hibákat. Amikor például a programmal tápláltuk a gépet, véletlenül másik játék jött be. Természetesen nem tudtuk mindjárt, hogy az ismeretlen játékot hogyan kell irányítani. A Dorks programban dinamitokat lehetett lerakni, és kikezdeni a tátogó szörnyetegekkel. Amikor végiggondoltam a játék tartalmát, már közel sem tetszett annyira. Kicselezni a másikat — még ha szörnyetegekről van szó, akkor is — úgy, hogy az nem veszi észre, mit csinálunk vele, egészen biztos hogy nem tisztességes.

De folytatom. Szóval, amikor megint betápláltuk a programunkat, ismét elhibáztuk, újból egészen más jött be. Ezt Killerwat-nek nevezik. A barátom kedvenc játéka lett. Bálnákat, struccokat és kísérleteket lehetett kilőni. Az az igazság, hogy nekem ez sem nyerte el a tetszésemet. Ami viszont igen, az az órás program volt. Összesen 40 órát találhattam el igen rövid idő alatt. Ez is volt a játék célja. Meg az, hogy a reflexeimet próbára tegye a feladat, ugyanis 1 órára 10 másodpercet kaptam. Ügyeskedni, figyelni kellett a játékban, s ha az ember elkelekótyáskodta a dolgokat, csúnyán veszíthetett. A végén már megszállottként próbáltam minél rövidebb idő alatt, a lehető legnagyobb pontszámot összegyűjteni. Kicsit be is csavarodtam. Egy ideig még éjszaka sem álmodtam másról, mint a különböző programokról.

Miután szünidőben számítógépeztünk, úgy szervezték, hogy délutánonként úszszunk. A testmozgás jó és hasznos, de sajnos elvon a monitor mellől. Így tehát, amikor sikerült, akkor kicseleztem az úszómestert és visszaosontam a számítógépterembe. Egy nagyobb sráccal haverok lettünk — délután ugyanis a felsősök használták a gépeket —, és megengedte, hogy tovább játsszam a gépen.

Mit mondjak még? Vidáman, klasszul telt a vakációmnak ez a része. Mindenképpen jobban, hasznosabban, mint azoknak a gyerekeknek, akik kulccsal a nyakukban ezalatt az utcán csavarogtak. Azóta meg hétvégeken folytatom a játékprogramok futtatását a szabadidőközpontban. Hétközben egy nagyfiú barátom programírásra tanít, továbbadja mindazt, amit az iskolában oktatnak neki.

Hát így indult Dávidnak az ismerkedés . . .

Krasznai Éva

Még egyszer az Enterprise-ról

Kopácsy Vilmos számítástechnikus olvasónk számos észrevételt tett Szinkronban az idővel című, az Enterprise gépet ismertető (1987/9.) cikkünk tartalmával kapcsolatban.

Kopácsy Vilmos az „A" Stúdió Szakcsoportnál dolgozik. Ők kaptak megbízást a Novotrade-től és a Centrum Áruházak Vállalattól az Enterprise géphez való szoftverek és perifériák fejlesztésére. Következzenek az észrevételek:

1. Az angol és német gép közötti különbség

Kétfajta gépet hoztak forgalomba: egy részük német, a többi angol klaviatúrás. Ennek az az oka, hogy a gépeket más-más forrásból szerezték be. (A gépek beszerzésének részletkérdéseiről nincs információnk, és nem is kutatjuk az üzelti hátteret.)

A felhasználó a kézikönyv 12-13. oldalán tájékoztatást kap a különféle gépekről is (igaz, röviden — a szerk.). A billentyűzetet tekintve a két géptípus között lényeges különbség nincs, viszont a gépekhez mellékelt (tartozó) BASIC interpreterek közötti különbség igen jelentős. Erről a : HELP parancs segítségével győződhetünk meg.

Angol BASIC	*Német BASIC*
WP	VSAVE
BASIC	VLOAD
	VDUMP
	BRD
	UK
	WP
	BASIC

Az angol cartridge esetén a rendszer csak a BASIC interpreterrel bővül, a német cartridge esetén öt többletparancs áll rendelkezésünkre. A VSAVE, VLOAD, VDUMP a videocsatornák kimentését, betöltését, illetve ezek kinyomtatását teszik lehetővé. A BRD és UK parancsok a klaviatúra és a hibaüzenetek nyelvének (angol, német) kiválasztására szolgálnak. A fejlesztés során észleltünk néhány paraméterezési különbséget a két BASIC egyes utasításai között. A két interpreter méretkülönbsége is jelentős: a német BASIC 32 k, amíg az angol 16 k.

2. A Z80-as mikroprocesszor órajele

A Z80—A típusú mikroprocesszor maximális órajele 4 MHz lehet.

3. A hanggenerátorral kapcsolatban

A hanggenerátor (csak) 3 hang- + 1 zajcsatornával rendelkezik. A hullámformát nem a felhasználó választja meg (ez mindig négyszögjel). A burkológörbét (ADSR) lehet programozni.

4. A PAUSE (HOLD) billentyű használata

A PAUSE (HOLD) billentyűvel a rendszer alatt futó programok végrehajtását függeszthetjük fel tetszőleges ideig — a gomb ismételt leütéséig.

5. A programmodulok

A programnyelvet tartalmazó (LISP, FORTH, ASMON) modulokat a gép bal oldalán lévő CARTRIDGE PORT-ra kell csatlakoztatni. PASCAL CARTRIDGE a levélíró szerint nem létezik és az eredeti szoftverajánlatban sem szerepel.

6. A RESET gomb és a ki-be kapcsoló

A gépen azért nem kapott helyet egy külön ki-be kapcsoló, mert az Enterprise „intelligens" RESET gombbal van ellátva. Ha egyszer nyomjuk meg, akkor egy melegindítás következik be, amely a memóriában tárolt adatokat, valamint a rendszerváltozókat nem változtatja meg. Kétszeri lenyomás hidegindításnak felel meg, ami a gép ki-be kapcsolásával egyenértékű.

7. Mit csinál a DAVE chip

A DAVE chip a következő funkciókat látja el:

— Hang generálása (3 hang- + 1 zajcsatorna)
— Memórialapozások vezérlése
— Megszakítások vezérlése
— A reset vezérlése
— 1 MHz-es órajel generálása
— Wait ciklusok generálása

Ezeken kívül a DAVE chip generálja a 4 Mbájt kezeléséhez szükséges A0—A15-ön felüli címvezetékeket (A16—A21, CART1, CART2), valamint a három I/0 strobe jelet a B5, B6, B7-es portok számára.

8. A LINE PARAMÉTER TABLE kezelése

Az LPT-vel definiálhat a felhasználó egy tetszőleges formátumú képernyőt. Az LPT ún. LINE PARAMETER BLOKK-okból áll, és a fizikai képernyő egy sorára vonatkozó adatokat tartalmazza. Így csinálhatjuk meg, hogy látszólag a keretre írunk, valójában pedig az LPT-t egészítettük ki 9 darab LPB-vel, amelyeket karakteres üzemmódba állítottunk. Az operációs rendszer az LPT-ben meghatározott képernyőt kezeli, a keret méretét pedig a LPB-ben definiáljuk. A képernyő megjelenítését a NICK chip vezérli.

9. A BASIC rövid ismertetése

Az Enterprise gépben BASIC interpreter van. A videocsatornák megjelenítésénél a felhasználó csak Y irányú paramétereket adhat meg, az X irányú elhelyezést a rendszer kezeli, illetve a videocsatornát méretétől függetlenül mindig középre helyezi. Az Enterprise minden eszközt csatornákon keresztül ér el.

A BASIC nyelvben ismert az ORD függvény, amelyet az Enterprise is használ. Az Enterprise BASIC interpretere a MICROSOFT 5.0 BASIC-jének változata. (Az MS—BASIC nem ismeri az ORD függvényt — a szerk.)

Az Enterprise BASIC interpretere felhasználja az EXOS nyújtotta grafikus lehetőségeket. Az EXOS — Expandable Operating System — bővíthető operációs rendszert jelent. A rendszert a gépi kódú programok az alábbiak szerint hívhatják:

RST Ø3ØH
DEFB xxh: ahol xx a meghívott funkció kódja.

A rendszerfunkciók kódjai megtalálhatók a felhasználói kézikönyvben. A paraméterátadások az A, BC, DE regiszterekben történnek.

10. A perifériákkal kapcsolatban

A lemezvezérlő kártya csak a SHUGART 401-es szabványának megfelelő egységeket képes kezelni. A lemezvezérlő operációs rendszert EXDOS-nak hívják.

A lemezvezérlő kártya képes 40 és 80 sávos, illetve egyoldalas és kétoldalas meghajtókat kezelni, de az adatrögzítés mindig MS—DOS formátumú. Ez annyit jelent, hogy az Enterprise-zal felírt lemezt az IBM PC-vel olvasni tudjuk és fordítva.

A perifériák áráról nem nyilatkozhatott a szerző, mivel azokat jelenleg fejlesztjük. (Reméljük, hogy igazodni fog a gép szerény árához — a szerk.)

A BÁV üzleteiben egy 80 karakteres monochrom monitor 8—15 ezer forintba kerül, a színes változat ára 25—40 ezer forint.

A printercsatlakozón keresztül csak Centronics felülettel rendelkező nyomtatókat illeszthetünk. A serial csatlakozón keresztül viszont bármilyen soros (RS232, RS422) szabványnak megfelelő perifériát lehet illeszteni (sornyomtató, modem stb.).

Reméljük, hogy ezek az észrevételek hasznosan egészítették ki a fent megjelölt cikkünket, és hozzájárulnak az Enterprise jobb megismeréséhez. Szívesen fogadnánk a géppel kapcsolatos további szakmai információkat.

Memóriaolvasó program

A program a gép memóriacímeinek értékét írja ki hexadecimális formában, valamint a cím tartalmának megfelelő karaktert.

RUN parancs után a program bekéri azt a két címet, ahonnan az olvasás kezdődjön, illetve ahol befejeződjön. Ezek után megindul az olvasás, melynek sebessége 6 cím/másodperc.

A kinyomtatott sor formája:

A sor első értékének címe (4 karakter + 1 szóköz),

8 db kétjegyű hexadecimális szám, a címek értékei a sor elején kiírt memóriacímtől kezdődően,

8 db negatív karakter, a címek értékeinek megfelelő karakterek.

Ha a memória értékéhez nem tartozik megjeleníthető karakter (például 14 = kisbetűváltó), akkor a program negatív pontot ír. A kiírás a CTRL gomb lenyomásával lassítható, az f7 gomb lenyomásával leállítható, illetve indítható. A program C64, C16 és Plus/4 gépen változtatás nélkül használható. VC20 esetén a következő változtatásokat kell beírnunk:

30 FOR I=E TO V STEP 4

60 FOR J=I TO I+3

A program a tisztán gépi kódban írt assembler fordítók futását nem zavarja.

STUMPHAUSER TAMÁS

```
10 INPUT E,V
20 IF E<0 OR E>65535 OR V<0 OR V>65535 THEN 10
30 FOR I=E TO V STEP 8
40 GOSUB 150
50 KK$=A$+" ": A$=""
60 FOR J=I TO I+7
70 A=PEEK(J) : GOSUB 280
80 KK$=KK$+A$+" " : A$=""
90 NEXT
100 KK$=KK$+"/■/"+K$ : K$=""
110 PRINT KK$
120 GET B$ : IF B$="/F7/"THEN GOSUB 430
130 NEXT
140 RUN
150 A(1)=INT(I/(1613)) : A=I-(A(1)*(1613))
160 A(2)=INT(A/(1612)) : A=A-(A(2)*(1612))
170 A(3)=INT(A/16) : A=A-(A(3)*16)
180 A(4)=A
190 FOR X=1 TO 4
200 IF A(X)<10 THEN A$(X)=MID$(STR$(A(X)),2,1)
210 IF A(X)>9 THEN A$(X)=CHR$(A(X)+55)
220 IF A$(X)="" THEN A$(X)="0"
230 NEXT
240 A$=""
250 FOR X=1 TO 4
260 A$=A$+A$(X)
270 NEXT : RETURN
280 B=A
290 A(1)=INT(A/16) : A=A-(A(1)*16)
300 A(2)=A
310 FOR X=1 TO 2
320 IF A(X)<10 THEN A$(X)=MID$(STR$(A(X)),2,1)
330 IF A(X)>9 THEN A$(X)=CHR$(A(X)+55)
340 IF A$(X)="" THEN A$(X)="0"
350 NEXT
360 A$=""
370 FOR X=1 TO 2
380 A$=A$+A$(X)
390 NEXT
400 IF B>31 AND B<128 THEN K$=K$+CHR$(B) : RETURN
410 IF B>159 AND B<192 THEN K$=K$+CHR$(B) : RETURN
420 K$=K$+"." :RETURN
430 GET B$ : IF B$="/F7/" THEN RETURN
440 GOTO 430
```

Felhasználói karakterek

ZX-SPECTRUM

A Spectrum SCREEN $ függvénye nem ismeri fel a felhasználói grafikus karaktereket, de erre egy rövid programmal megtanítható. A 23606 és a 23607 címeken található értékek a karakterkészlet kezdőcímét határozzák meg. Nincs más dolgunk, mint ide az UDG terület címét írni, pontosabban annál 256-tal kevesebbet. Ezután a SCREEN $ a felhasználói grafikus karakterek között keresi a képernyőn levő jelet. Végrehajtás után ne feledkezzünk meg az eredeti karakterkészlet címének visszaírásáról:

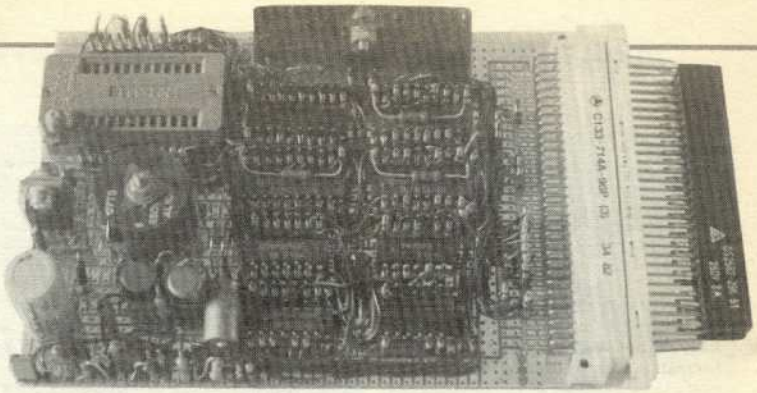
POKE 23606,0: POKE 23607,60

Természetesen a két „üzemmód" együtt is használható, például így:

```
10 IF SCREEN $ (x,y)="" THEN GO
   TO 40
20 LET a$=SCREEN $ (x,y)
30 GO TO 80
40 LET s=USR "a"
50 POKE 23606,s-256*INT (s/256)
60 POKE 23607,INT (s/256)-1
65 LET a$=SCREEN $ (x,y)
70 POKE 23606,0: POKE 23607,60
80 PRINT AT 0,0; a$
```

Az x sorszámra, y oszlopszámra írt normál vagy felhasználói grafikus karaktert a program felismeri, az a$ változóba helyezi, majd a 0,0 pozícióra kiírja.

MOLNÁR LÁSZLÓ



2—4 kbájtos EPROM-a Spectrumhoz

Az amatőr gyakorlatban legtöbbször a 2 és 4 k-s EPROM-ok égetése fordul elő. Az alábbi „toldalék" a 2716, 2732 és 2732A típusú EPROM-ok égetésére szolgál.

A BASIC sorok bebillentyűzése és magnóra mentése után (RUN 9999) a program a későbbi betöltéseknél automatikusan indul. Figyelem! Az EPROM-égetőt a Spectrummal *csak kikapcsolt állapotban* szabad csatlakoztatni, majd bekapcsolás után lehet a BASIC programot betölteni. Minden más esetben előre nem látható károk keletkezhetnek számítógépünkben!

A program leírása

170	a RAMTOP beállítása
180	a C üzemmód beállítása
190—250	a gépi kód betöltése
260	2 k üzemmód beállítása
270	alaphelyzet-beállítás
290—390	a menü kiírása a képernyőre
400—500	billentyűfigyelés
510—540	magnóra mentés
550—570	betöltés magnóról
580—640	EPROM-beolvasás
650—700	égetés-ellenőrzés
710—790	a kiválasztott EPROM égetése
800—950	programozás a memóriában
960—1040	2 k rendszerváltozók beállítása
1050—1130	4 k rendszerváltozók beállítása
1140—1330	a gépi kódú rutinok területe
9999—	magnóra mentés és automatikus indítás

A betöltés után a program automatikusan indul és a képernyőn az ADATHALMAZ TÖLTÉSE A MEMÓRIÁBA szöveg jelenik meg. Ezalatt a Spectrum a DATA sorokban elhelyezett gépi kódokat egy adott memóriaterületre tölti. A töltés végeztével megjelenik a menü:

MAGNÓRA MENTÉS (S)
BEOLVASÁS MAGNÓRÓL (L)
EPROM—BEOLVASÁS (B)
ÉGETÉS (E)
ELLENŐRZÉS (V)
PROGRAMOZÁS (P)
EPROM TÍPUSA (2—4)

A program működése

Magnóra mentés. Az S gomb hatására a program magnetofonra menti az aktuális memóriaterületet (E000—E7FF vagy E000—EFFF). Ha más elfoglaltság miatt nem tudjuk teletölteni az adott memóriaterületet, akkor az addig megírt részt elmenthetjük magnetofonra, hogy másnap ott folytathassuk, ahol abbahagytuk.

Beolvasás magnóról. Az L gomb választása esetén a program betölti a magnetofonról a már előzőleg kimentett, 2 vagy 4 kbájt hosszúságú adathalmazt.

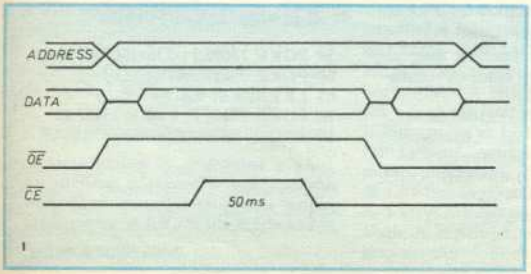
EPROM-beolvasás. Ha a B gombot nyomjuk le, megjelenik a következő szöveg: HELYEZD AZ EPROMOT A FOGLALATBA ÉS GOMBNYOMÁS. Az EPROM behelyezése és ismételt gombnyomás után az EPROM tartalma beíródik az E000-ás címtől.

Égetés. Az E gomb hatására a program megkérdezi a kezelőt: ÉGETNI AKARSZ(I), védekezve a véletlen gombnyomás ellen. Az I gomb lenyomására a memóriában tárolt információ beíródik az EPROM-ba. Minden más gomb hatására a program visszatér a menühöz. Az égetés végét sípszó jelzi, majd a menü jelentkezik. Az égetés idején a megszakításengedélyezés le van tiltva, hogy az égetés alatt ne következhessen be megszakítás az ULA felől.

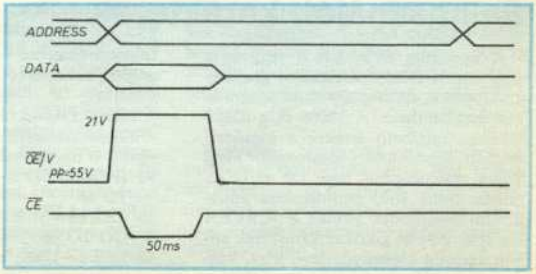
Ellenőrzés. A V gomb lenyomására a program összehasonlítja az EPROM és a memória tartalmát. Hiba esetén kiírja a hibás bájtok számát (a maximális érték 255).

Programozás. A P gombbal a memóriaterületet lehet szerkeszteni, rendezni, javítani. Kilépés: P+ENTER, léptetés előre:

1. ábra. A 2716 égetési diagramja



2. ábra. A 2732 égetési diagramja



ENTER, ugrás előre 15 bájtot: Ø+ENTER, ugrás előre 255 bájtot: 1+ENTER. Az EØØØ-ás címtől progamozhatunk gépi kódban Ø—F-ig, más gomb hatására a program hibaüzenettel leáll. Indítása: GOTO 28Ø.

EPROM típusa. Ha a 2—4 gombot választjuk, a képernyő aljára kiírja az aktuális EPROM típusát (2 vagy 4 bájt), és átírja a használt rendszerváltozókat is. A program nem tesz különbséget a 2732 és a 2732A között. Erre a pogramozónak kell figyelnie, hogy a 2732-re a 25 V, a 2732A-ra 21 V égetőfeszültség jusson. A tápegységnél e célra egy kapcsoló szolgál.

égető

A BASIC program viszonylag egyszerű, ezért némi tanulmányozás után bárki kedve szerint egyszerűsítheti vagy bonyolíthatja.

A rendszerváltozók területe

DØØØ DØØ1	az inicializált EPROM kezdőcíme beolvasáskor
DØØ2	az EPROM típusa
DØØ3	a program hossza
DØØ4	ellenőrzéskor a hibabit helye

A gépi kódok kezdete DØ1Ø = 53264
A memóriaterület kezdete EØØØ = 57344

A bemeneti és kimeneti címek

FF1F kimeneti címen az égetendő bájt jelenik meg

FF3F kimeneti címen az alsó (A_0A_7) bájt jelenik meg

FF5F kimeneti címen a felső címbájtok (A_8—A_{11}), valamint a $\overline{CE}$, $\overline{OE}$,a 2732 égetőfeszültség engedélyezése és az adattároló engedélyező jele

FF9F bemeneti címen olvasható az EPROM tartalma

FFBF kimeneti címen engedélyezi a 2716 21-es lábára az 5 V-ot

FFDF kimeneti címen engedélyezi a 2716 21-es lábára az égetőfeszültséget (25 V)

FF7F kimeneti címen tiltható a feszültség engedélyezése 2732 esetén

Az olvasási folyamat

FF5F címre küldjük a felső címeket és beállítjuk a $\overline{CE}$ $\overline{OE}$ jeleket,

FF3F címre küldjük az alsó címeket,

FF9F címről az akkumulátorba töltjük az adatot, az akkumulátorból kiküldjük egy memóriacímre, amit a HL értéke határoz meg. Ez a művelet ciklusba van szervezve és addig tart, amíg a "Program hossza" rendszerváltozó értéke egyenlő a HL értékével.

Az írás folyamata

FF5F címre küldjük a felső címeket és beállítjuk a $\overline{CE}$ $\overline{OE}$ jeleket,

FF3F címre kiküldjük az alsó címeket,

FF1F címre kiküldjük az égetendő bájtot,

CALL égető rutin.

Ez a folyamat is ciklusba van szervezve és addig tart, amíg a "Program hossza" rendszerváltozó értéke egyenlő a HL regiszter értékével.

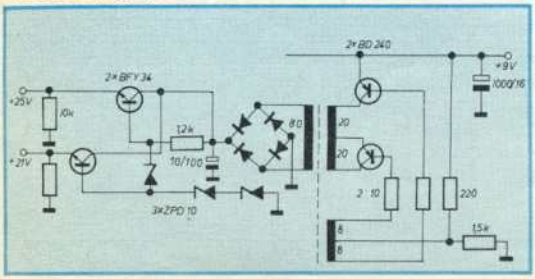
Fontos! Bekapcsoláskor és kikapcsolás előtt nem lehet EPROM az égetőben, továbbá nem ajánlott a bekapcsolt Spectrumról lehúzni vagy arra felhelyezni az EPROM-égetőt!

Az összeszerelés

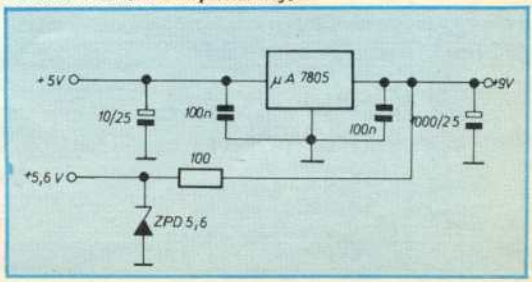
Az összeszerelést célszerű univerzális NYÁK-lemezen végezni. Az IC-ket az EPROM kivételével érdemes beforrasztani, mert a jó foglalat drága, az olcsóbbal pedig sok bosszúságot okozhatunk magunknak. Az összehuzalozásnál a Wire-Wrap technológiát használtam, de természetesen forrasztani is lehet. Vigyázzunk a zárlatra és az elkötésre, ellenőrizzük többször is, mert nagy károkat okozhatunk. Az égetőt DS 868 FC 64 típusú Euro-csatlakozóval láttam el. A Spectrum és az égető közé egy közdarabot tettem, amely a gép felől egy levágott DS 2582—296 51-es csatlakozó, az égető felé pedig egy DS 868 EC 64 Euro. Azért választottam ezt a megoldást, mert az Euro-csatlakozó olcsóbb is és könnyebben beszerezhető. Ezután ha bővíteni akarjuk rendszerünket, már csak Euro-csatlakozóra van szükségünk. Az összeforrasztás előtt csípjük ki az Euro-csatlakozó A 1, 2, 31, 32 és C 1, 2, 31, 32 lábait, mert ezeket nem használjuk. A számítógépből az alábbi jelvezetékekre van szükség:

A Spectrum kivezetései	Az Euro kivezetései
DØ = 6A	C25
D1 = 7A	C24
D2 = 8A	C23
D3 = 11A	C2Ø
D4 = 12A	C19
D5 = 10A	C21
D6 = 9A	C22
D7 = 3A	C28
$\overline{IORQ}$ = 17A	C14
$\overline{RD}$ = 18A	C13
M1 = 24A	C7
A5 = 23B	A8
A6 = 22B	A9
A7 = 21B	A10
"O" = 7B	A24

3. ábra. A tápegység

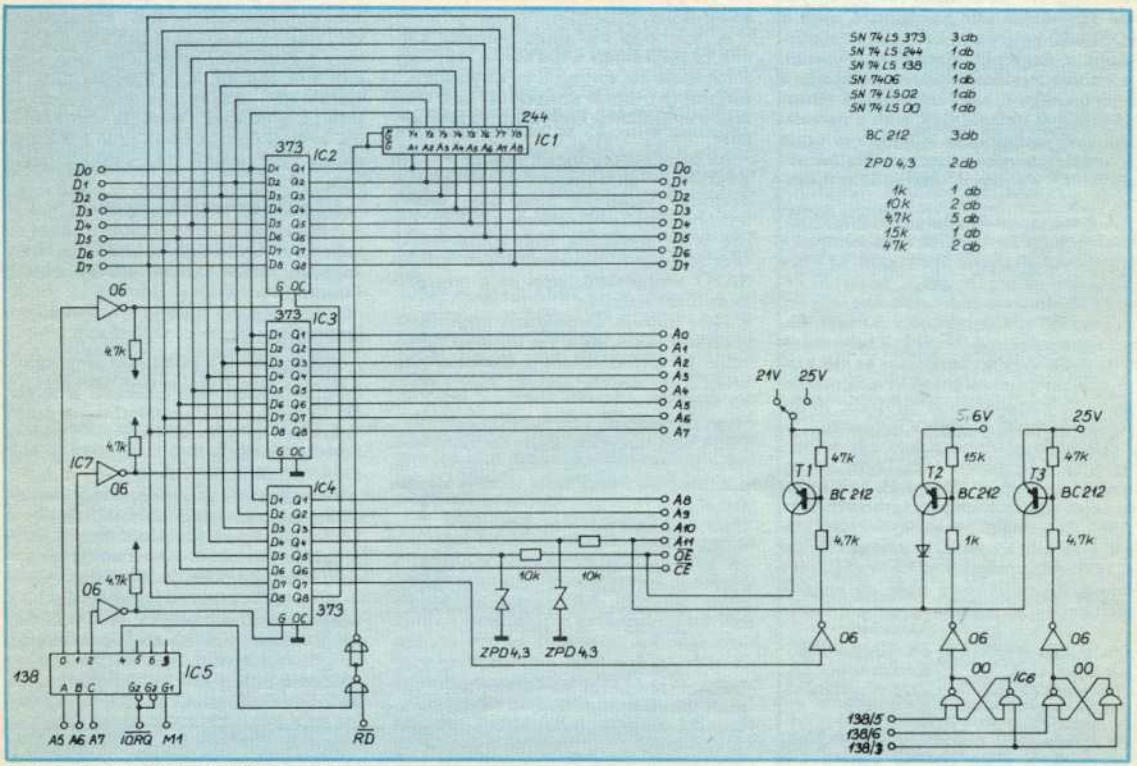


4. ábra. A transzverter kapcsolási rajza



```
100 REM
110 REM +++++++++++++++++
120 REM +               +
130 REM + HARAY GYONGYI +
140 REM +               +
150 REM +++++++++++++++++
160 REM
170 CLEAR 53247: REM    CFFF-H
180 POKE 23658,8: REM NAGYBETU
190 PRINT AT 5,0;"ADATHALMAZ TO
LTESE A MEMORIABA"
200 LET B=53264: REM    D010-H
210 FOR A=1 TO (20+34+6+16+37+8
+77+70)
220 READ C
230 POKE B,C
240 LET B=B+1
250 NEXT A
260 GO SUB 960
270 RANDOMIZE USR 53264
280 CLS
290 PLOT 30,165: DRAW 200,0: DR
AW 0,-116: DRAW -200,0: DRAW 0,1
16
300 PRINT AT 2,5;"MAGNORA     ME
NTES     (S)"
310 PRINT AT 4,5;"BEOLVASAS  MA
GNOROL  (L)"
320 PRINT AT 6,5;"EPROM BEOLVAS
AS      (B)"
330 PRINT AT 8,5;"EGETES
        (E)"
340 PRINT AT 10,5;"ELLENORZES
        (V)"
350 PRINT AT 12,5;"PROGRAMOZAS
        (P)"
360 PRINT AT 14,5;"EPROM TIPUSA
    (2-4)"
370 BRIGHT 1: FLASH 1
380 IF H1=2 THEN PRINT AT 20,12
;"2-KBYTE": BRIGHT 0: FLASH 0: G
O TO 400
390 PRINT AT 20,12;"4-KBYTE": F
LASH 0: BRIGHT 0
400 IF INKEY$<>"" THEN GO TO 40
0
410 IF INKEY$="" THEN GO TO 410
420 IF INKEY$="S" THEN GO TO 51
0
430 IF INKEY$="L" THEN GO TO 55
0
440 IF INKEY$="B" THEN GO TO 58
0
450 IF INKEY$="2" THEN GO SUB 9
60
460 IF INKEY$="4" THEN GO SUB 1
050
470 IF INKEY$="E" THEN GO TO 71
0
480 IF INKEY$="P" THEN GO TO 80
0
490 IF INKEY$="V" THEN GO TO 65
0
500 GO TO 370
510 CLS
520 IF H1=2 THEN SAVE "2-KBYTE"
CODE 57344,2048: GO TO 280
530 SAVE "4-KBYTE"CODE 57344,40
96
540 GO TO 280
550 CLS
560 LOAD ""CODE 57344
570 GO TO 280
580 CLS
590 RANDOMIZE USR 53264
600 PRINT AT 8,0;"HELYEZD AZ EP
ROMOT A FOGLALAIBA"
610 PRINT AT 10,8;"ES GOMBNYOMA
S !"
620 PAUSE 0
630 RANDOMIZE USR 53284
640 GO TO 280
650 CLS
660 RANDOMIZE USR 53340
670 IF PEEK 53252>=1 THEN PRINT
 AT 8,5;"EGETES ";PEEK 53252;"
HIBAVAL !!": POKE 53252,0: GO TO
 690
680 PRINT AT 8,4;"AZ EGETETT PR
OGRAM JO !!"
690 PAUSE 0
700 GO TO 280
710 CLS
720 PRINT AT 6,5;"EGETNI AKARSZ
?    (I)"
730 IF INKEY$<>"" THEN GO TO 73
0
740 IF INKEY$="" THEN GO TO 740
750 IF INKEY$<>"I" THEN GO TO 2
80
755 CLS : PRINT AT 10,10;"EGETE
S !"
760 RANDOMIZE USR 53385
770 RANDOMIZE USR 53264
780 BEEP 1,1
790 GO TO 280
800 CLS
810 LET CIM=57344
820 DEF FN A(A$,B)=CODE A$(B)-4
8-7*(CODE A$(B)>57)
830 DEF FN C(A$)=16*FN A(A$,1)+
FN A(A$,2)
840 DEF FN G$(F)=CHR$ (F+48+7*(
F>9))
850 DEF FN H$(E)=FN G$(INT (E/1
6))+FN G$(E-16*INT (E/16))
860 DIM A$(2)
870 PRINT FN H$(PEEK CIM);TAB 7
;FN H$(INT (CIM/256));FN H$(CIM-
256*INT (CIM/256));
880 INPUT A$
890 IF A$(1)="P" THEN GO TO 280
900 IF A$(1)="O" THEN LET CIM=C
IM+15: GO TO 930
910 IF A$(1)="I" THEN LET CIM=C
IM+255: GO TO 930
920 IF A$(1)<>CHR$ 32 THEN POKE
 CIM,FN C(A$)
930 PRINT TAB 16;FN H$(PEEK CIM
)
940 LET CIM=CIM+1
950 GO TO 870
960 BRIGHT 1: FLASH 1
970 PRINT AT 20,12;"2-KBYTE": F
LASH 0: BRIGHT 0
980 LET H1=2
990 POKE 53248,0
1000 POKE 53249,136
1010 POKE 53250,2
1020 POKE 53251,232
1030 POKE 53252,0
1040 RETURN
1050 FLASH 1: BRIGHT 1
1060 PRINT AT 20,12;"4-KBYTE": F
LASH 0: BRIGHT 0
1070 LET H1=4
1080 POKE 53248,0
1090 POKE 53249,128
1100 POKE 53250,4
1110 POKE 53251,240
1120 POKE 53252,0
1130 RETURN
1140 REM ALAPHELYZET
1150 DATA 62,0,1,127,255,237,121
,1,63,255,237,121,62,176,1,95,25
5,237,121,201
1160 REM BEOLVASAS
1170 DATA 243,33,0,224,237,91,0,
208,58,2,208,254,2,204,70,208,20
5,76,208,119,35,19,58,3,208,188,
194,62,208,205,16,208,251,201
1180 REM OUTPUT  (2-K)
1190 DATA 1,191,255,237,121,201
1200 REM INPUT OUTPUT
1210 DATA 1,63,255,237,89,1,95,2
55,237,81,1,159,255,237,120,201
1220 REM ELLENORZES
1230 DATA 243,58,2,208,254,2,204
,70,208,33,0,224,237,91,0,208,20
5,76,208,190,196,129,208,35,19,5
```

```
8,3,208,188,194,108,208,205,16,2
08,251,201
1240 REM HIBA SZAMLALASA
1250 DATA 58,4,208,60,50,4,208,2
01
1260 REM 2-K EGETO
1270 DATA 243,58,2,208,254,2,194
,214,208,33,0,224,17,0,24,1,95,2
55,237,81,1,63,255,237,89,126,1
31,255,237,121,205,181,208,19,35
,62,232,188,194,152,208,251,201
1280 REM EGETO RUTIN
1290 DATA 1,223,255,237,121,62,3
2,178,1,95,255,237,121,1,0,29,11
,121,176,194,197,208,1,95,255,23
7,81,1,127,255,237,121,201
1300 REM 4-K EGETO
1310 DATA 33,0,224,17,0,48,1,95,
255,237,81,1,63,255,237,89,126,1
31,255,237,121,205,249,208,19,3
5,62,240,188,194,220,208,251,201
1320 REM EGETO RUTIN
1330 DATA 122,246,64,1,95,255,23
7,121,230,223,237,121,245,1,0,29
,11,121,176,194,9,209,241,246,32
,01,95,255,237,121,230,191,237,1
21,201
9999 SAVE "HARAY" LINE 170
```



5. ábra. Az EPROM-égető elvi kapcsolási rajza

"O" = 6B A25
9V = 4B A27

Az I_{c1} olvasáskor az EPROM egy adott cím szerinti tartalmát az adatbuszra teszi.

Az I_{c2} íráskor az égetendő adatot az EPROM adatbuszára teszi.

Az I_{c3} feladata az alsó címbitek tárolása (A_0—A_7).

Az I_{c4} feladata a felső címbitek (A_3—A_{11}) az $\overline{OE}$, $\overline{CE}$, a 2732 égetőfeszültsége, valamint az I_{c3} engedélyezése.

Az I_{c5} végzi a címzéseket.

Az I_{c6} biztosítja íráskor a 25 V, olvasáskor pedig az 5 V feszültséget a 2716 részére.

A Zener-diódák megakadályozzák az égetőfeszültségnek az I_{c4} kimeneteire jutását.

A T_1 kapcsolja a 2732 égetőfeszültségét, a T_2 kapcsolja a 2716 5 V-ját olvasáskor, a T_3 kapcsolja a 2716 25 V-ját égetéskor.

A tápegység egy lehetséges megoldása, hogy a Spectrum 9 V-os tápjából transzverter segítségével állítjuk elő az égetőfeszültségeket. A vas-AL 100 fazékmag mérete 22 × 17 mm.

A transzverter adatai:

2 × 20 menet 0,5 mm rézhuzal
2 × 8 menet 0,2 mm rézhuzal
1 × 80 menet 0,2 mm rézhuzal.

Az 5 V-os tápot áteresztővel, az 5,6 V-ot Zener-diódával állítjuk elő. A diódákat tűrésük miatt válogatni kell:

$U_{z1} = 25 \pm 1$ V
$U_{z2} = 21 \pm 1$ V

Hűteni csak a 7805-ös stabilizátort kell; erre a célra megfelel egy 60 × 40 × 1 mm-es alulemez. A beméréshez jól szolgál egy csővoltmérő és egy EMG 1555 vagy EMG 1567 oszcilloszkóp.

HARAY GYÖNGYI

Integrált szoftver

Fájlok összerendelése

A cikksorozatunk előző részében közölt program az adatok vagy szövegfájlok egyesítését, összefésülését végzi, tehát a SZÖVEGSZERKESZTŐ és a POSTÁZÓ „termékeivel" dolgozik.

A POSTÁZÓ a címjegyzéket tartalmazó fájlt állítja elő. Ez azonban ebben a formában közvetlenül nem használható, ezért a POSTÁZÓ program <E> opciójával elkészítjük a megfelelő, kivonatolt változatot. A kivonat természetesen tartalmazhatja a teljes rekordot is, ha ez szükséges. A választás menüből történik, úgy, mint a nyomtatási opció esetében. Az új fájlban az egyes rekordok szövegformátumban tárolódnak; így például a kódmező nem jut át az új fájlba.

A menü utolsó kérdése: FÁJLNÉV? Válaszként azt a nevet adjuk meg, amelyen a kivonatolt fájlt tárolni akarjuk. Jó, ha a név emlékeztet az eredeti fájlra, amelyből készült. A program kivonatoláshoz egy új, még szabad területre tekeri a szalagot. Ehhez természetesen előbb be kell helyeznünk a magnetofonba egy kazettát és be kell kapcsolnunk a magnetofont. A kivonatolás folyamatát a képernyőn is láthatjuk. A fájl végén a program visszatér a főmenübe.

A kivonat készítésénél még egy dologra felhivjuk a figyelmet. A POSTÁZÓ programmal a szöveges címjegyzék bevitelekor az egyes rekordok csupa nagybetűvel is írhatók. Ez azonban néhány nyomtatási képnél — például a levélnél — nem szerencsés. A POSTÁZÓ program <T> opciójával a szöveg visszaalakítható nagy- és kisbetűs formára:

DR. NAGY VAZUL	Dr. Nagy Vazul
KARIKA U. 6.	Karika u. 6.
7777 NAGYFALU	7777 Nagyfalu

Az átalakításnál minden mező első betűje nagybetű, a többi pozíció kisbetű lesz. A konverzió lassú folyamat, ezért csak a kivonat készítésénél (<E> opció) célszerű alkalmazni. Ahhoz, hogy pontos működését megismerjük, érdemes kipróbálni.

Mielőtt a szövegfájlok összefésülését elindítjuk, a SZÖVEGSZERKESZTŐ-vel elő kell állítani a szöveget, úgy, mintha normál szövegszerkesztő üzemmódban lennénk, csak a név és cím helyére a /m vagy /ms jelölést írjuk be. Az összefésülő program a /m-et a POSTÁZÓ által készített kivonatolt név- és címfájlban található szöveggel helyettesíti. A /ms jelöléskor a név és cím mellett automatikusan generálódik a megszólítás is, például: Tisztelt Dr. Nagy!

Mivel a SZÖVEGSZERKESZTŐ-ben használt /m vagy /ms jelölés csak egy sort vesz igénybe, szemben az összefésülésnél szükséges 4—7 sorral, ügyeljünk arra, hogy elég hely legyen a papíron!

A /m opció sort emel, beolvassa a címet, és kinyomtatja a cím három vagy négy sorát. Ha a program a címben üres sort talál, kihagyja azt.

A /ms opció sort emel, beolvassa a címet, és kinyomtatja a cím három vagy négy sorát, majd két sort emel és kinyomtatja a megszólítást. Ennél az opciónál nem lehet csak a megszólítást kinyomtatni a levél elejére!

Az összefésülő program minden esetben a POSTÁZÓ által készített fájl kivonatával dolgozik, függetlenül attól, hogy esetleg minden címet felhasználunk a fájlból. Ennek az a magyarázata, hogy a levél megkívánt formátuma rendszerint elért a POSTÁZÓ programétól, ezért ez a megoldás látszott célszerűnek.

Nem történik automatikus megszólításgenerálás akkor, ha a /m opcióval dolgozunk. Ez esetben írhatunk általános megszólítást is: Kedves Vásárló! vagy Kedves Kolléga! stb.

A /ms opciónál a megszólításgenerálás a POSTÁZÓ program által használt első, második és harmadik mező (megszólítás, cím, rang stb., családnév, utónév) alapján alakul ki:

Címe, rangja	Családnév, utónév	Megszólítás
Dr.	Nagy Vazul	Kedves Dr. Nagy
Prof.	Szabó Alajos	Kedves Prof. Szabó
	Szivárvány Áruház	Tisztelt Vásárlónk

Láthatjuk, hogy egy kis ötletességgel változatos formátumokat állíthatunk elő.

A SZÖVEGSZERKESZTŐ-ben a kívánt levélformátumnak megfelelően a /sn (=soremelés n-szer) utasítás megelőzheti a /m vagy /ms utasításokat. Ezeknek az utólagos szerkesztéseknek az eredményét célszerű nyomtatásban is ellenőrizni, mert az összefésülő programban szöveg- vagy formátummódosításokat nem végezhetünk; ezt csak a SZÖVEGSZERKESZTŐ-vel vagy a POSTÁZÓ-val tehetjük meg. Gyakori hiba például, hogy a címzés miatt az eredeti levél néhány sora átkerül a másik oldalra, és ezt csak a második vagy harmadik levélnél vesszük észre. Ezért a többpéldányos nyomtatásnál mindig ellenőrizzük a formátumot!

A munka menete a következő:

1. Készítsük el a szövegfájlt (például levelet) a SZÖVEGSZERKESZTŐ-n és /m vagy /ms jelölést írjunk a cím és a megszólítás helyére.

2. Rögzítsük a szövegfájlt a kazettán a szokásos módon.

3. Készítsük el a POSTÁZÓ programmal a kívánt címjegyzékkivonatot egy másik kazettán.

4. Töltsük be és futtassuk a POSTÁZÓ programot, adjuk meg a fájlneveket, ahogyan a program menüje kívánja.

Ne vegyük ki a címjegyzékfájl-kazettát addig, amíg az összefésülés tart! A címrekordokat hármas blokkban olvassa be a program. Amint az összefésülés megtörtént, a programkazetta automatikusan leáll.

Ha a SZÖVEGSZERKESZTŐ-nél a /K opciót választottuk (dátum, név és cím utólagos beírása nyomtatáskor), az összefésülő program ezt figyelembe veszi, de ilyenkor a nyomtató mellett kell ülni és várni a gépelés kezdetét adó jelre. /K opciónál egyszer-

```
?.
[BH]   (V)   3350
.....A.........B.........C.....
A              Havi kiadások
B Megnev.      összege      Százalék
C-------------------------------------
D Lakás          1250.00        10.13
E Közmü           845.00         6.85
F Hitel          2400.00        19.46
G Ruházat        1338.00        10.84
H élelem      >  3350.00        27.15
I Közlek         1375.00        11.14
J Egyéb           680.00         5.51
K Takarék        1100.00         8.92
L =====================================
M TOTAL         12338.00       100.00
```

re csak egy címsor vagy mondat írható be, és ez azonnal nyomtatásban is megjelenik.

TÁBLÁZATKEZELŐ (előzetes)

Ezek után már csak egyetlen program maradt hátra: a TÁBLÁZATKEZELŐ, amely a legbonyolultabb, legösszetettebb ebben a sorozatban. Elöljáróban tekintsük át feladatát és működését.

A táblázatkezelő programok elsősorban a könyvelésekben a pénzügyi mérlegek és az iparban az anyagmérlegek készítésénél használt nagyméretű adattáblázatok feldolgozását könnyítik meg. Ezek az előre nyomott táblázatok arra szolgálnak, hogy a számok (például mérési eredmények) az összefüggéseknek megfelelően szervezett formában kerüljenek feldolgozásra. A tartalom egyértelmű azonosítására a sorokat és oszlopokat címkével látják el. Egy-egy speciális célú táblázat létrehozása tulajdonképpen próbálkozásokon alapuló hosszú és fáradságos munka eredménye. A sorokat, oszlopokat sokszor kell újraírni, újraszámolni. Az utólagos igényeket elég nehéz figyelembe venni anélkül, hogy az egész táblázatot újra kelljen számolni, szerkeszteni.

A számítógépes táblázatkezelő programok megkönnyítik a táblázatkészítést; elvégzik a szükséges számításokat, kinyomtatják az eredményeket, és az elkészült táblázatot későbbi felhasználás céljára eltárolják. Az ismertetendő programban megtalálhatók a „nagygépes" táblázatkezelők legfontosabb tulajdonságai, de természetesen a gép korlátaiból adódóan lényegesen kisebb méretekben.

A program ismertetése során néhány új fogalom is szóba kerül majd. A SOR és OSZLOP elnevezést sokan ismerik: az előbbi számok, kifejezések, címkék vízszintes elrendezése, az utóbbi pedig ennek függőleges megfelelője. A bemutatandó BASIC program 26 sor és 26 oszlop egyidejű kezelésére képes. Az így kialakuló háló vagy más néven rács 676 helyére lehet számot vagy címkét írni. Ezeket a helyeket a továbbiakban mezőknek nevezzük. Minden sort és oszlopot az angol ábécé egy-egy betűje jelöl, A-tól Z-ig. Minden mezőnek van egy címe, amely az oszlopot és a sort jelölő betűből áll. Például a táblázat bal felső sarkában lévő mező címe AA lesz, a bal alsó sarok AZ, a jobb felső sarok ZA stb.: az első helyen álló betű az OSZLOP-ot, a második helyen álló a SOR-t jelöli ki.

A táblázatkezelő program Dragonra készült verziójával 3 oszlop és 13 sor jelenik meg egyszerre a képernyőn. A mindenkori ablakokat a bal oldalon és a táblázat tetején található címek, betűk jelölik ki.

A képernyő tetején két sor az ún. munkaterület, amelyen keresztül beírhatók az információk a táblázatba. A mezőkbe számértékek, számítási formulák vagy címkék kerülhetnek.

Nézzük meg az *ábrán* látható példát. A *számértékek* (VALUES), amelyeket a képernyőn (V) betű jelöl, a számítógép gyártója által megadott értékeken belül bármelyek lehetnek. Jelen esetben 9 jegyre számolunk. Az értéket a kalkulátor üzemmódban közvetlenül be lehet írni, de egy számítási formula eredményeként kapott érték is kerülhet a megfelelő mezőbe.

A *kifejezés* (FORMULA) — a képernyőn (F) jelöli — számok és/vagy mezőnevek sorozata, amelyeket aritmetikai jelek választanak el. Például: +AA/AB*AC+200. Itt szigorúan érvényesül a balról jobbra precedenciaszabály, tehát: az AA mező tartalmát elosztja az AB mező tartalmával, az eredményt megszorozza az AC mező tartalmával és hozzáad 200-at. Zárójelek használhatók a jobb áttekinthetőség kedvéért, de ezeket a program a számolás során figyelmen kívül hagyja! A számítási kifejezés eredménye mindig egy szám lesz, amelyet a program a kijelölt mezőben megjelenít, kivéve ha azon a címen egy címke takarja az értéket.

Egy *címkét* (LABEL) — a képernyőn (L) — mindig üres hellyel kell kezdeni; a program erről ismeri fel, hogy címkét kezdtünk beírni. Ha egy szám elé szóközt teszünk, a szám címkeként fog szerepelni. Miután a program a címkét eltárolta, eltávolítja a szóközt. Minden oszlop 10 karakter széles, de egy címke „átcsordulhat" a szomszédos mezőbe is. A címke egyszerűen eltávolítható: mindössze egy szóközt kell beírni a megfelelő pozícióra.

A képernyőn megjelenítéskor fölérendeltséget tapasztalunk: az azonos mezőre megadott címke mindig fedi az értéket. Ezáltal munkaterületeket tehetünk el például aláhúzások, magyarázó szövegek mögé. Nulla számértéket a program nem jelenít meg, kivéve, ha a nulla egy számítás eredményeként került a cellába. A számítási kifejezések nem jelennek meg a táblázatban.

Mint már említettük, a képernyő felső két sora munkaterület, amelyen keresztül információkat vihetünk be a táblázatba. Az első sor adatbevitel kérésére utaló jelet (?) és a feladat kijelölését tartalmazza. A ! jel például az újraszámolást jelenti, a : pedig a nyomtatást a képernyőre.

A második sorban a kurzorpozíción található mező legfontosabb információi íródnak ki: ez rendszerint egy kifejezés vagy számérték, de lehet címke is.

A valóságban a táblázat három szint mélységű. A számértékek, a kifejezések és a címkék egymástól elválasztva tárolódnak, ami által a program meglehetősen rugalmassá válik.

Legközelebb részletesen ismertetjük a programot és használatát, majd néhány példával szemléltetjük a működését.

GALINA FERENC

µINFORM

A tartalomleírások az alábbi folyóiratokban megjelent programlistákról készültek:

A folyóirat neve	Kódja
64'er Magazin	64er
Chip Magazin	chip
Commodore Horizons	coho
Commodore Microcomputers	comi
Compute!	cute
Computer Persönlich	pers
Happy Computer	happ
hc - Mein Home-Computer	hc
mc - Zeitschrift	mc
Run /USA/	run
Sinclair User	sinc
Your Sinclair	ysin

A tartalomleíró szóláncokat permutáltuk, a szólánc variánsokat pedig alfabetikusan rendeztük.

A tartalomleírás egy szóláncból áll, majd a listában ezt követi a forrás megjelölése a folyóirat azonosítójával, a megjelenés dátumával és a cikk előkereséséhez a kezdő oldalszám és a terjedelem megadásával. A mellékelt lista értelmezéséhez még az alábbiakat kell tudni. A tartalomleírás szóláncában elsőként a téma átfogó megnevezése, utána a számítógéptípus(ok), ezt követően a szűkebben jelölt tartalom meghatározása szerepel, majd esetlegesen néhány, a közleményt minősítő adat (például : cikksorozat).

A forráshely karaktersorozatát nyíl vezeti be, melyet a / jelig a folyóiratok azonosítója, a két / jel között az évszám, folyóiratszám és kötőjellel a kezdő oldalszám követi, a végén pedig a közlemény teljes oldalterjedelme áll.

A folyóiratok a SZÁMALK szakkonyvtárában (Budapest XI., Szakasits Á. út 68. Nyitva: 8-tól fél 5-ig. Tel.: 853-111/251) is follelhetők. A kiválasztott anyagról másolat rendelhető az alábbi formában:

SZÁMALK Szakkonyvtára
Budapest, 112. Pf.: 146. 1502

Megrendelem a Mikroszámítógép Magazin 1987/ sz. alapján a következő folyóirat-oldal-másolatokat:

Kód: _____ Példányszám: _____
Kód: _____ Példányszám: _____
Kód: _____ Példányszám: _____

A megrendeléshez csatolom az oldalankénti 8,- Ft-os szolgáltatási díj befizetését igazoló csekkszelvényt.
Dátum, név, pontos cím.

PROGRAMLISTA
animacio commodore 128 grafikai magic lantern ~comi 86.03-104 C
PROGRAMLISTA
apple iic parameter beallitasi sarokba illesztes ~hc 86.10-77 4
PROGRAMLISTA
assembler programozas cikksorozat commodore 64 jatekprogram keszitei ~happ 86.11-70 4
PROGRAMLISTA
assembler programozas cikksorozat commodore 64 128 atteres basicrol ~64er 86.10-152 5

μINFORM

PROGRAMLISTA
atari 800 xl programlemeredes listaresetmodositas automatikus torlesher
->happ/86.10-104/1
PROGRAMLISTA
atari xl/xe 256-karakteres 256-karakteres futoszoveg generalasa
->happ/86.10-84/1
PROGRAMLISTA
atari xl/xe grafika "dia-show"
->happ/86.11-65/2
PROGRAMLISTA
barkacsolas commodore 64 fotozas mesterseges intelligencia kameran szereplo expozicios ido ellenorzese
->64er/86.12-46/2
PROGRAMLISTA
barkacsolas commodore 64 illeszto centronics nyomtatas rezzitezi utmutato
->64er/86.12-57/8
PROGRAMLISTA
cikksorozat commodore 128 vdc regiszterek kezelese ->64er/86.11-88/3
PROGRAMLISTA
cikksorozat commodore 64 jatekprogram kezzitese "moon base" mint peldaprogram ->comi/86.09-84/4
PROGRAMLISTA
cikksorozat commodore 64/128 grafika matrixszamitas transzformacio
->64er/86.12-146/5
PROGRAMLISTA
commodore 128 cp/m formatumterezes
->64er/86.10-98/2
PROGRAMLISTA
commodore 128 jatekprogram "sprite puzzle" ->comi/86.09-112/3
PROGRAMLISTA
commodore 16/64/plus/4 "drawstick" vonalrajzolas botkormannyal
->run/86.11-90/2
PROGRAMLISTA
commodore 64 "joystick-mauz"
->hc/86.10-44/1
PROGRAMLISTA
commodore 64 autostart rutin a 1541 hez ->happ/86.11-78/3
PROGRAMLISTA
commodore 64 basic precommpiler
->64er/86.12-68/5
PROGRAMLISTA
commodore 64 basic programvaltozato kereszthasonlitasa
->run/86.12-106/3
PROGRAMLISTA
commodore 64 belso-ora assembler programozasa ->happ/86.10-59/4
PROGRAMLISTA
commodore 64 giga-cad grafika nyomtatas ->64er/86.12-97/2
PROGRAMLISTA
commodore 64 giga-cad grafika rajzgep(1520) vezerles ->64er/86.11-74/2
PROGRAMLISTA
commodore 64 gorgeto szubrutinok
->run/86.10-11/1
PROGRAMLISTA
commodore 64 grafika "64'grafix" hires gepi kod/8 utasitas
->run/86.10-68/4
PROGRAMLISTA
commodore 64 grafika monitorkep kinyomtatasa ->run/86.11-68/2
PROGRAMLISTA
commodore 64 grafika sprite kezelese atkapcsolhato 38/40 oszlopos abrazolas ->64er/86.12-91/2
PROGRAMLISTA
commodore 64 grafika tengelykoruli forgatas ->64er/86.11-54/7
PROGRAMLISTA
commodore 64 i/o gyorsito lemezegyseg(1541) muveletgyorsitas basic-rom atlati ram hasznalata
->64er/86.12-54/3
PROGRAMLISTA
commodore 64 jatekprogram "cave raid"
->happ/86.10-63/3
PROGRAMLISTA
commodore 64 jatekprogram korokelet-poke keresesi modszerek
->happ/86.11-155/4
PROGRAMLISTA
commodore 64 mutatogeneralas menurhoz pozicionalas botkormannyal
->run/86.10-78/2
PROGRAMLISTA
commodore 64 nyomtatas(mps801) haromszinu masolat indigopapirral
->64er/86.10-105/4
PROGRAMLISTA
commodore 64 nyomtatas(mps801) kepmeret csokkentes ->64er/86.10-95/1
PROGRAMLISTA
commodore 64 oktatas(szamitastechnika) logikai operatorok kezelese
->run/86.11-90/2
PROGRAMLISTA
commodore 64 prodisc kiegeszites a c 86.06)-ban kozolt listahoz
->64er/86.11-89/1
PROGRAMLISTA
commodore 64 programozasi tippek haladoknak ->64er/86.10-81/4
PROGRAMLISTA
commodore 64 programozasi tippek haladoknak ->64er/86.11-76/3
PROGRAMLISTA
commodore 64 programtorles nelkuli katalogusbehivas 108 byte-os parancsrutin ->happ/86.10-72/1
PROGRAMLISTA
commodore 64 tukorkeprajzolas
->run/86.10-86/2
PROGRAMLISTA
commodore 64/128 egermurodesi eljaras apcsolasi rajz ->64er/86.11-44/4
PROGRAMLISTA
commodore 64/128 nyomtatas(epson) kepmeret csokkentes ->64er/86.10-94/1
PROGRAMLISTA
commodore 64/128 oktatas(morse-abc) jelkonvertalas adott vevo uzemmodhoz
->run/86.11-74/4
PROGRAMLISTA
commodore 64/vc20 jatekprogram oktatas(matematika) "math scramble"
->run/86.11-82/3
PROGRAMLISTA
commodore 64/vc20 programmodulok osszekapcsolasa ->run/86.10-88/3
PROGRAMLISTA
commodore 64/vc20 visszamentes scratch utan ->comi/86.09-82/2
PROGRAMLISTA
helyesbites (86.08)76
->hc/86.10-50/1
PROGRAMLISTA
helyesbites (86.09)datassette-segedprogramfram-killer ->happ/86.11-130/1
PROGRAMLISTA
sakk atari xl/xe commodore 64 ket-szamitogepes jatek ->happ/86.11-61/3
PROGRAMLISTA
sinclair spectrum decimalis-lebegopontos atalakitas(automatikus)
->happ/86.11-70/1
PROGRAMLISTA
sinclair spectrum egerszimulacio botkormannyal ->happ/86.10-86/2
PROGRAMLISTA
sinclair spectrum gorgeto rutin
->sinc/86.12-125/1
PROGRAMLISTA
sinclair spectrum grafika
->hc/86.10-46/4
PROGRAMLISTA
sinclair spectrum jatekprogram keszites high-score lista kiiratasa
->hc/86.10-45/2
PROGRAMLISTA
sinclair spectrum jatekprogram "axehero" ->sinc/86.11-106/2
PROGRAMLISTA
sinclair spectrum jatekprogram "back to font" ->sinc/86.12-129/2
PROGRAMLISTA
sinclair spectrum jatekprogram "boggle" ->sinc/86.11-105/2
PROGRAMLISTA
sinclair spectrum jatekprogram "forest of the long shadows"
->sinc/86.10-100/4
PROGRAMLISTA
sinclair spectrum jatekprogram "mouse al" ->sinc/86.11-100/1
PROGRAMLISTA
sinclair spectrum jatekprogram "pro ace word" ->sinc/86.12-132/1
PROGRAMLISTA
sinclair spectrum jatekprogram "sos distress" ->sinc/86.12-129/1
PROGRAMLISTA
sinclair spectrum jatekprogram "tree bullder" ->sinc/86.12-130/2
PROGRAMLISTA
sinclair spectrum tasword gyors alfabetikus rendezes ->happ/86.10-88/1
PROGRAMLISTA
sinclair spectrum tasword pascal programozas file-formatum atalakitas
->happ/86.11-68/1
PROGRAMLISTA
sport commodore 64 "bundesliga-manager" ->hc/86.10-35/9
PROGRAMLISTA
startexter szovegfeldolgozas commodore 64 monitorkep formalas es lemezeriras ->64er/86.11-172/1
PROGRAMLISTA
szervizbarkacsolas cikksorozat lemezegyseg(1541) hibakeresese fejpozicionalas ->64er/86.12-48/3
PROGRAMLISTA
szovegfeldolgozas commodore 128 (run script 128) ->run/86.12-61/20
PROGRAMLISTA
szovegfeldolgozas commodore 64 master-text formatum-atalakito idegen filekhoz ->64er/86.11-98/2
PROGRAMLISTA
szovegfeldolgozas commodore 64 nyomtatas ascii konverzio karakterkeszlet modositas ->64er/86.10-28/4
PROGRAMLISTA
szovegfeldolgozas vizawrite cikksorozat commodore 64 helyesiras ellenorzes ->64er/86.11-65/6
PROGRAMLISTA
terminal uzemmod commodore 128 (mini term 128) ->comi/86.09-110/2
PROGRAMLISTA
vizawriter cikksorozat commodore 64 giga-cad grafika szoveg-kep szerkesztes ->64er/86.10-179/6
PROGRAMLISTA
zene barkacsolas commodore 64 hangdigitalizalo keszitesi utmutato
->64er/86.10-65/7
PROGRAMLISTA
zene commodore 64 (zound monitor) alkalmazasi utmutato
->64er/86.10-53/9
PROGRAMLISTA
zene commodore 64 hypra-basic sid-programozas poke nelkul
->64er/86.11-85/2

Az Olivetti nem hibázik!

Jobb kettő, mint egy sem

Az olaszországi Olivetti cég is magáévá tette e szólás alapigazságát, és megkezdte egy új gépcsalád gyártását. A katonai, űrhajózási, pénzügyi stb. rendszerekben alapvető követelmény a minél nagyobb biztonság és a folyamatos üzem. Az új gépcsalád, amely ezen elv szerint működik, az OLIVETTI CPS 32 nevet viseli. A CPS a Continuous Processing System (folyamatos feldolgozási rendszer) rövidítése.

A rendszer alapkoncepciója nem új: duplázzunk meg minden hardveregységet, így egy rendszer helyett kettő fog egyszerre dolgozni. Ha valamelyik egységben hiba jelentkezik, a másik folytatni tudja a munkát. Az elképzelés azonban mindeddig mégsem valósult meg, mert a megduplázott áramkörökkel működő gépek igen költségesek lettek volna. Mégis, az elmúlt tíz évben jelentős változás történt ezen a téren.

A hardverelemek olcsósága, méretük és energiaigényük csökkenése lehetővé tette az elgondolás megvalósítását. A szoftverköltségek emelkedése viszont a különálló gépek összekapcsolása ellen szólt. Korábban próbálkoztak ugyanis különálló, hagyományos nagygépek „párhuzamosításával", ami az éppen futó programnak megfelelő ellenőrzőprogramot is igényelt.

A CPS tehát egy gépen belül duplázza meg az áramköri elemeket, és az ellenőrzést is saját beépített monitorprogramjával végzi.

Hasonlítsuk össze a hagyományos rendszereket a CPS-sel! Ha egy hagyományos rendszerben valamilyen hiba keletkezett, azt nem mindig lehetett azonnal észlelni. A feldolgozás tovább folytatódott, gyakran hibás eredményeket szolgáltatva.

A CPS 32 rendszer minden egysége tartalmaz egy hardvermonitort, amely a feldolgozott adatok folyamatos ellenőrzését végzi. Az önellenőrzés másodpercenként nyolcmilliószor megy végbe, így az esetleges hibát a rendszer gyakorlatilag keletkezésének pillanatában észleli. Ez igen jelentős előnyökkel jár:

— a feldolgozott adatok garantáltan hibamentesek,

— nincs szükség hosszadalmas adathiba-feltárásra és újrafeldolgozásra,

— elhagyható a bonyolult áramköri hibakeresés, mivel a rendszer jelzi, hogy hol történt a meghibásodás,

— a hibajavítás időtartamára nem kell a rendszert leállítani.

A CPS 32 család az FT 200, az XA 400 és az XA 600 típusú gépekből áll. Minden gép virtuális operációs rendszer (Virtual Operating System) alatt dolgozik.

Mivel az elemek duplázottak, ugyanazon a feladaton egyszerre két rendszer dolgozik. Ha egy alkatrész meghibásodik, a párja a felhasználó zavarása nélkül folytatja munkáját. A feldolgozás sebessége ekkor sem csökken.

Az előfordult hibáról külön jelzést ad. A hibás egységet a rajta kigyulladó piros színű világítódióda jelzi. A hibás egység a feldolgozás alatt, annak megszakítása nélkül kicserélhető. Az új egység behelyezését észleli a rendszer, az egységet automatikusan ellenőrzi, majd újra beállítja működési feltételeit, és a feldolgozás már ismét a dupla rendszeren fut anélkül, hogy a felhasználó ezalatt egy másodperc kiesést is tapasztalt volna.

ZSOM BÉLA

ADOK–VESZEK–CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (60 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,-Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

Angol nyelvvizsga-előkészítő szótanító program C64-re. 11 témakörben 1400 szót tanít, kiejtéssel. 2-3-szor gyorsabban tanulhat, mint szótárból. Ára magánszemélyeknek és közületeknek floppyn 3600 forint. Kívánságra részletes tájékoztatót küldök. Tóth András számítástechnika, Post Restant, 1364 Budapest 4.

C64-re angol nyelvet oktató programokat - összesen öt lemezoldalnyi anyagot - eladnék vagy elcserélnék. Egy oldal ára 100 Ft + a lemez ára (amennyiben szükséges). Eladó továbbá a Flight Deck nevű játékprogram, ára 200 Ft. Illés István, Cikó, Perczel M. u. 44. 7161

C64 programkazettákat eladók. Kazettánként 50-60 program. Egy kazetta 500 Ft (+ utánvétel). Tóth József, Hajdúböszörmény, Tyereskova u. 22. 4220

C64 magnóval, joystickkel, programkazettákkal és bő szakirodalommal eladó. Irányár: 28 000,- Ft. Solyák Zsolt, Nyíregyháza, Május 1. tér 11. 4400

Commodore 1901-es színes monitor (új) PAL és RGB csatlakozóval olcsón eladó. Tokaji Zoltán, Budapest, Petőfi S. u. 5. 1052 vagy: 18-77-48 telefonüzenet.

Commodore 64 programok titkosítását floppyn vállalom magánszemélyeknek és közületeknek. Minden másoláson szemben biztos védelem, igen nehéz visszafejthetőség. Levélben részletesen tájékoztatom. Báldi András, Post Restant, 1364 Budapest 4.

Programok C Plus/4 és C16 számítógépekre: FORTH nyelv, Turbo file kazettás fájlkezelő, Disk system BASIC és assembler fejlesztőprogram, Copy 226 lemez/szalagos másoló, C64-C Plus/4 turbo, C64-C Plus/4 rutin és rendszerváltozó összehasonlító táblázat, Profiass C Plus/4 kétmenetes assembly fordító. A programok ára 180-600 forint, kazettán, utánvéttel, részletes kezelési utasítással. Kérésre ismertetőt küldök. Pelsőczi Gyula, Szilasliget, Ady Endre u. 36. 2145

C64 és C128-tulajdonosok figyelem! Junoszty televíziót videomonitorrá alakítok (kép+hang), az eredeti funkciók megtartása mellett. C128 esetén 80 oszlopos megjelenítésre is alkalmas. Páll Miklós, Budapest, II. Rákóczi Ferenc u. 349. k 7-10 C lépcső. 1214

SONY személyi számítógép (MSX rendszerű) - Z80A, 64 k RAM, 448 k ROM, 256x192-es felbontású grafika, 3 hangcsatorna, személyi adatbank -, 2 db joystickkel, cartridge-zsel, dokumentációval eladó. Tel.: 26/49-051

Videoton TV-Computer olcsón eladó. Tel.: 647-887

ZX81 (64 kbájt) sok programmal és joystickkel eladó. Árajánlatot levélben kérek. Vigh György, Budapest, Margit u. 8. 1165

ZX-Spectrum (48 k) olcsón eladó. Vadon Zoltán, Budapest, Törcsvár u. 28. 1112. Tel.: 669-539

VESZEK

Vennék üzemképtelen VC/VIC 20, HT 10807, ABC 80, Primo, ZX80, ZX81, TIMEX 1000, TIMEX 2068, C16, AIRCOMP 16 számítógépet és tartozékaikat. (Darabok is érdekelnek.) Árajánlatokat a következő címre kérem: Maróti Gyula, Bordány, Zákányszéki u. 22. 6795

CSERÉLEK

C16 és C Plus/4 programokat cserélek. Surányi Péter, Miskolc, Kórosi Cs. u. 7. 3527

C16 és C Plus/4 programokat cserélek. Gerencsér Tamás, Zalaegerszeg, Landorhegyi út 18 B. III/9. 8900

C64-re készült programokat cserélek (elsősorban kazettán, de lemezen is). Hajdu László, Budapest, Szigeti József u. 17. V/32. 1041

C64 programokat kazettán cserélek. Bódai József, Pápa, Fő tér 14. 8500

Commodore 64-es számítógépre játékprogramokat cserélek kazettán. Választékot listával a következő címre: Banyák Mihály, Mosonmagyaróvár, Széchenyi u. 19. II/8. 9200

ZX-Spectrum felhasználói és színvonalas játékprogramokat cserélek. Szarka Endre, Veszprém, Stadion u. 22. 8200

Milyen feladatok oldhatók meg számítógéppel?

Képes-e egy számítógép például árkot ásni, zoknit mosni vagy egy nehéz ládát felcipelni az első emeletről a harmadikra? Általában úgy tartjuk: nem, de a válasz attól függ, hogyan *definiáljuk* a számítógépet, hol húzzuk meg a határt a számítógép és a környezet között.

Másfelől: miért nem nevezünk egy egyszerű zsebkalkulátort *számító*gépnek, csak (megkülönböztetésül) *számoló*gépnek? Általánosabban: mi az a *minimális tudás*, mellyel egy eszköznek rendelkezni kell ahhoz, hogy a számítógép nevet kiérdemelje?

A számítógép meghatározásával kapcsolatos kérdések láthatóan két csoportra oszthatók: az egyik csoport *felülről*, a másik *alulról* próbálja behatárolni a gépet.

Meghatározásaink, fogalmaink, elnevezéseink

Gyakran önkényesnek tűnhetnek — mögöttük azonban rendszerint mély társadalmi *tapasztalat* húzódik meg. Andor Csaba könyvének (Jel-Kultúra-Kommunikáció, Gondolat, 1980.) példáját idézve: megtehetjük-e, „hogy a lovak egy részét — mondjuk a kancákat — a zsiráfok osztályával egyesítve, ezentúl ezt az osztályt jelöljük egy meghatározott szóval? A választ nem is olyan könnyű megadni, mert egyfelől valóban nincs *elvi* akadálya egy ilyen — meglehetősen „bizarr" — osztályozásnak, másrészt viszont semmiképpen sem véletlen, hogy egyetlen olyan nyelvet sem ismerünk, amelyben külön szó jelölné az állatoknak ezt a keverék osztályát . . . Ennek az az oka, hogy a nyelvhasználat során a *ló*-val jelölt osztályra gyakran, míg a „kanca vagy zsiráf" osztályra úgyszólván sohasem történik utalás, így célszerűbb az előbbi osztályt egyetlen jellel, míg az utóbbit szükség esetén valamilyen nyelvi leírással helyettesíteni. Persze az sem véletlen, hogy az egyik osztályt lényegesen gyakrabban kell megnevezni, mint a másikat: végső soron a *társadalmi gyakorlat* határozza meg, hogy az egyes természetes nyelvek milyen megkülönböztetési elvek szerint osztják fel a világot.

Ha tehát a „kanca vagy zsiráf" osztályra egyetlen nyelvben sincs külön szó, akkor ezt úgy értelmezhetjük, hogy ez az osztály egyetlen közösség életében sem játszik fontos szerepet."

A számítógép meghatározása mögött is nyilván az húzódik meg, hogy a közösség mit tekint a számítógépek lényegének."

A minimális tudás

A számítógépektől megkövetelt *minimális tudás* problémája érdekes módon a számítógépek *előtti* időkre nyúlik vissza, és összefügg azzal a filozófusok és matematikusok által feszegetett problémával, hogy mi a

> valóság, a bizonyíthatóság
> és a megoldhatóság

viszonya. A történelmi vonulatot többek között olyan nagy nevek fémjelzik, mint Arisztotelész (i.e. 384-322), Hilbert (1862-1943), Gödel (1906-1978), Turing (1912-1954) — mint ahogy ezt egy 1986-ban készített és Magyarországon még a nagyközönség elé nem került videofilm is összefoglalta (Dr. Kurt Gödel: Ein Mathematischer Mytos. P. Weibel és W. Schimanovich, 1986.) Reméljük, hogy ezt a filmet valamikor a szélesebb hazai közönség is láthatja majd.

A brünni születésű osztrák Gödel volt az, aki 1930-ban többek között bebizonyította, hogy egyetlen formális rendszer sem lehet tökéletesen zárt; a valóság erősebb, mint a *bebizonyíthatóság*. Vannak olyan feladatok, melyek egy formális rendszeren belül bizonyíthatóan *megoldhatatlanok*.

Számunkra különösen érdekes például az, hogy egy programmal nem lehet kideríteni, hogy egy másik programban vannak-e végtelen ciklusok.

Az angol Turingot is a megoldhatóság problémái foglalkoztathatták, amikor 1936-ban kigondolt egy igen egyszerű szerkezetet — egy számítógépet — (lásd az *1. ábrát*), amelyről be lehetett bizonyítani (a bizonyítás jóval meghaladja e cikk kereteit), hogy vele *minden más, szimbólumokkal műveleteket végző gép működése elvileg tökéletesen szimulálható.* Azaz ez az egyszerű gép (megfelelően programozva!) képes — a sebességtől eltekintve — *tökéletesen* úgy viselkedni, mint bármilyen más szimbólumfeldolgozó gép. (Megjegyezzük, hogy cikkünkben nem térünk ki arra, hogyan lehetett egy ilyen egyszerű felépítésű géppel [majdnem] minden, szimbólumokra visszavezethető feladatot megoldani. Az ábrával csak azt kívánjuk érzékeltetni, hogy milyen egyszerű [?] gépről van szó.) Bizonyítható továbbá, hogy az a feladat, mely ezzel a Turing-féle géppel elvileg nem oldható meg, az *semmilyen* más számítógéppel sem lesz megoldható.

Amikor azt mondjuk, hogy a számítógépek közös, lényegi vonása az, hogy (megfelelően programozva őket!) minden olyan feladat megoldására képesek, melyekre a Turing-féle gép képes, akkor meghatároztuk azt a *minimumot*, mely ahhoz kell, hogy egy szerkezetnek a számítógép nevet adhassuk.

Természetesen sok olyan szimbólumfeldolgozó gép van, mely *nem* képes minden olyan feladatot megoldani, melyet a Turing-féle gép *elvileg* képes lenne. Ilyenek lehetnek például az automata háztartási gépekbe, egyes szerszámgépekbe szerelt elektronikus vezérlő, számítógép*szerű* célszerkezetek, de ilyenek például a már korábban említett zsebkalkulátorok is.

Szimbólumok, mintázatok, tükröztetés

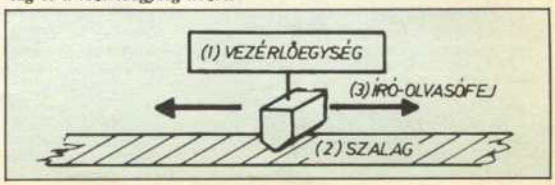
Láttuk, hogy az elvárható minimális „tudást" illetően a számítógépek általános célú *szimbólum*feldolgozó eszközökként definiálhatók. A *szimbólum* (a jelkép) szót a köznyelv is használja: olyan ábrát, tárgyat stb.-t szoktunk így megnevezni, mely valami más (fogalom, dolog, személy stb.) helyett áll, valami mást „ábrázol". A mindennapi életben a szimbólumot és azt, amit ábrázol, összekapcsolhatja az emberi képzelet (a gondolattársítás), de összekapcsolhatja vagy a társítást megerősítheti pontos leírás, hagyomány, akár jogszabály is (lásd például az állami jelvényekre vonatkozókat!).

Számítógépekkel dolgozva a megoldandó feladatok világát *lekép*ezzük a gép által kezelhető fizikai *mintázatok* (a gép alkotóelemei állapotában megtestesülő) világára. A *fizikai* jelzővel itt arra utalunk, hogy ma a gépek alkotóelemei — a feladatok világát tükröző állapotaikat-állapotváltozásaikat tekintve — a *fizika* (konkrétabban: az elektronika, a villamosságtan és nem például a kémia vagy a biológia) törvényei szerint működnek.

A gép képes a *mintázatokat* összehasonlítani, illetve képes azokkal más műveleteket is végezni. A gép által kezelhető szimbólumok készletét (a gép ábécéjét) éppen az határozza meg, hogy a gép ezt az alapvető összehasonlítást milyen mintázatokon képes elvégezni. Ezt a tudást persze a gép tervezője építette be a gépbe.

A feladatok világának *tükröztetése* a szimbólumok világára nem új dolog azoknak sem, akik számítógépet még sohasem láttak — legfeljebb nem tudatosult bennük a művelet ezen aspektusa, akárcsak az „Úrhatnám polgár" Jourdain urában az, hogy egész életében „prózában beszélt". Például amikor az iskolai feladatainkat „lefordítottuk" a matematikai egyenletek nyelvére, és ezeket az

1. ábra. A Turing-féle gép
(1) A vezérlőegység véges számú állapot felvételére képes
(2) A szalag diszkrét részekre (négyszögekre) van osztva, melyek mindegyike egy véges készletből vett, egyetlen szimbólum tárolására képes
(3) Az író-olvasó fej a szalag menetén lépésenként (négyzetenként) képes előre/hátra mozogni és információt továbbítani (oda/vissza) a szalag és a vezérlőegység között



is és eltünödtünk azon, hogy lényegét tekintve mit is nevezünk számítógépnek. Fontos ez? Megyünk-e ezzel valamire? Szoktunk-e gondolkodni például azon, hogy mi egy nadrág lényege? Felvesszük, levesszük, kimossuk, eldobjuk, satöbbi és kész. Joggal furcsállnánk, ha valaki elkezdené: a nadrág olyan háromlyukú ruhadarab, melyen az egyik lyuk nagyobb és ellenkező irányú, mint a másik kettő . . ., satöbbi. A számítógépekkel is így vagyunk: bemegyünk a boltba, vásárolunk egyet, majd bekapcsoljuk, használjuk definíció nélkül is. A kérdésnek nincs közvetlen gyakorlati jelentősége, inkább csak ürügy arra, hogy a számítógép-fogalmat körbejárjuk.

egyenleteket leírtuk a füzetünkbe, szintén ilyesmit tettünk. Amikor például leírjuk azt a szót egy levélben, hogy „kutya" és közben egy létező, élő, konkrét kutyára gondolunk, akkor a valóságos kutyát az ábécé betűiből összeállított karaktersorra *tükröztettük*.

Hol a „felső határ"?

Szokás a „turingi" (a lényegi, a szükséges minimumot tartalmazó) résszel elektromosan összekötött szerkezetek együttesét *konfigurációnak* nevezni, a teljes konfigurációt pedig a hagyományos és a ma tipikus gépeknél szokás (és lehet!) ún. *központi egységre* (CPU-ra) és ún. *perifériális egységekre* (röviden csak: perifériákra) osztani. A perifériák tipikusan szimbólumok be- és kivitelére, átvitelére vagy ún. *háttértárolásra* szolgálnak (lásd a *2. ábrát*). De hol a CPU határa, hol kezdődnek a perifériák? A számítógépek (az ún. konfiguráció) felosztása központi egységre és perifériákra egyre nehezebb lesz.

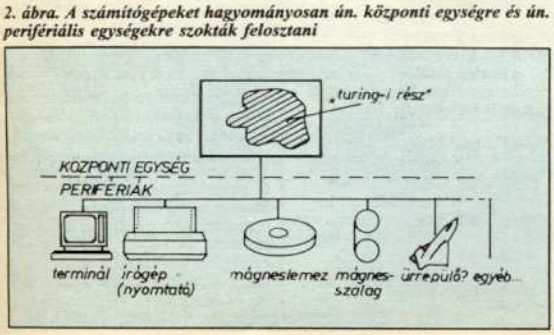
A CPU-hoz kapcsolódó eszközök halmaza elvileg korlátlan. Tekinthető-e például egy beépített számítógéppel vezérelt űrrepülőgép a számítógép különleges perifériájának? Elvileg nem lenne akadálya egy ilyen felfogásnak sem. Gyakorlati érzékünk persze tiltakozna ez ellen.

Ha a számítógépekhez a velük elvileg megoldható feladatok felől közelítünk és a számítógép-fogalomba beleértjük a perifériákat *is*, akkor nem jutunk messzire a lényeg megfogásában, mert (legalábbis elvben) nem fogunk találni olyan (megoldható!) feladatot, mely az ilyen tágan értelmezett géppel ne lenne megoldható. Ilyen értelmezéssel ugyanúgy nem jutnánk közelebb a lényeghez, mint ha a vasút lényegét az alapján próbálnánk megragadni, hogy „ki mindenki utazhat vasúton" (cipész, postás, polgármester, tökmagárus).

Visszautalva a „kanca-zsiráf" példára, a számítógép lényegét kutatva akkor járunk el célszerűen, ha azt a *turingi*, a minimálisan szükséges részt *is* tartalmazó CPU környékén keressük.

A gyakorlatban általában abból indulunk ki, hogy mindaz, ami a lényegi (a „turingi") résszel egy mechanikai egységet (egy szerelvényt) alkot, azt nevezik központi egységnek. A központi egységek az elméleti minimumnál általában persze sokkal többet tartalmaznak, mert más dolog kigondolni egy gépet egy fontos tétel bebizonyításához, ahogy ezt Turing is tette (ez legyen lehetőleg minél egyszerűbb, az se baj, ha nagyon lassú, hiszen úgysem építik meg, csak gondolatkísérleteket végeznek vele), és megint más olyan gépeket tervezni, amelyekkel feladatok sokaságát kell majd gyakorlati követelményeket is kielégítő módon (gyorsan, olcsón stb.) megoldani.

2. ábra. A számítógépeket hagyományosan ún. központi egységre és ún. perifériális egységekre szokták felosztani



A „felső határt" (azt, hogy mit ne értsünk a lényegi részbe a számítógép környezetéből) keresve meg kell elégednünk tehát azzal, hogy gyakorlati szempontok alapján leválasztjuk a CPU-ról a perifériák tarka sokaságát, tudva, hogy a CPU-fogalom is képlékeny, és az is elképzelhető, hogy hosszabb távon értelmét is veszti.

Mire jutottunk?

Nem nagyon sokra. Megállapítottuk, hogy a számítógépek *lényegüket tekintve* általános célú szimbólumfeldolgozó szerkezetek, melyekbe legalább a „turingi tudás" be van építve. Azt, hogy ez pontosan micsoda, itt nem mondtuk meg, csak egy ábrával utaltunk a Turing-féle gépre. Szó volt arról, hogy ma a számítógépeket — az ún. *konfigurációkat* — CPU-ra és perifériákra szokás felosztani. Tudjuk, hogy az elhatárolás esetleges, önkényes. Csak önkényesen tudtuk elhatárolni a gépeket a külvilágtól, a környezetüktől is.

Röviden körüljártuk a szimbólumok, a mintázatok és a tükröztetés fogalmait, és visszahúzódtunk a szimbólumfeldolgozás világába.

Kikerültük azt a kényes kérdést, hogy a gép—szimbólum—művelet „hármas" csak *együtt* értelmezhető igazán, bár aki figyelmes volt, az felfigyelhetett arra, hogy például a szimbólumok készletét éppen az határozza meg, hogy a gép milyen mintázatok megkülönböztetésére képes az *összehasonlítási művelet* kapcsán.

Ami az „általánoscélúságot" illeti, ez nem jelenti azt, hogy a számítógépekkel *minden* olyan feladat megoldható, mely szimbólumokkal végezhető műveletekre vezethető vissza. Vannak ugyanis megoldhatatlan feladatok. Beleütköztünk a „gödeli problémába". Azt, hogy mely feladatok nem megoldhatók, azt éppen Turing-féle gépek segítségével vizsgálják.

Ezek után miben maradjunk? Először is abban, amiből kiindultunk: a kérdésnek *nincs* közvetlen, gyakorlati jelentősége. A mindennapi életben nem okoz gondot a pontos definíció hiánya. A számítógépeinkkel naponta megoldandó feladatok sokasága pedig *megoldható!* A megoldatlan (vagy ma annak látszó) problémák elsősorban a filozófusok, a matematikusok számára érdekesek. Az átlagfelhasználó ezekkel *nem* találkozik. Egyébként is megtanultunk olyan világban élni, tájékozódni, melyben — anélkül, hogy tudtuk volna vagy számunkra bebizonyították volna — homályosan éreztük (vagy nem törődtünk vele), hogy fogalmaink nem alkotnak tökéletesen zárt rendszert. Talán ösztöneinkben érezzük, amit Gödel már 1930-ban bebizonyított, hogy a „valóság erősebb a bizonyíthatóságnál", hogy minden formális rendszeren belül vannak, lehetnek magán a rendszeren belül bizonyíthatatlan tételek, sejtések. Ha a számítógéphez nem a velük elvileg megoldható feladatok oldaláról kívántunk volna közelíteni, talán nem is botlottunk volna ilyen elvi problémába. Azoknak, akiket a problémakör jobban érdekel, a remélhetőleg előbb-utóbb szélesebb körben hozzáférhető, idézett videofelvétel mellett Péter Rózsa: Játék a végtelennel című (Budapest, 1957.) könyvét ajánljuk.

Befejezésül még annyit: a számítógépek, ahogy lekerülnek a tervező asztaláról vagy a gyári futószalagról, elvileg (láttuk, hogy ez mit jelent) *univerzálisak*, azaz a gyakorlati életben (majdnem!) minden feladat megoldására programozhatók. Egy adott pillanatban egy konkrét gép azonban *csak* azoknak a feladatoknak a megoldására képes, melyekre felkészítették:

— melyek megoldására a megfelelő programot elkészítették (megtervezték, kipróbálták) és

— „futásra kész" állapotban betöltötték a gépbe.

A megoldható feladatok köre egy-egy konkrét esetben tehát *rögzített*, és ezt a kört a *tervező* határozza meg. (Olyan értelemben természetesen a felhasználó is, hogy betölti a tervező által előzetesen elkészített programot.)

Messze vezetne annak a kérdésnek a boncolgatása, hogy mi van akkor, ha gépünk egy kiválasztott pontjára hangszórót vagy színes monitort kapcsolunk és így (előre *nem* tervezett módon!) érdekes — netán „művészi" — képeket vagy „zenét" *is* nyerünk, miközben a gép előre eltervezett módon éppen valamilyen matematikai problémát old meg. Hagyjuk ezt a kérdést is a filozófusokra!

—KE—

A játékfa és kiértékelése

Legutóbb az iteratív keresési eljárással foglalkoztunk, amelynek lényege, hogy az algoritmus nemcsak a végső csomópontban értékeli a játékfát, hanem a közepén is, és ennek alapján rendezi. Így elértük, hogy nagy valószínűséggel a legjobb változatot elemezzük elsőként, és ennek eredményeként az alfa-béta algoritmus levágja a fa következő leveleit. Ezzel sok időt nyerhetünk, amit még több változat átszámítására fordíthatunk, növelve programunk játékerejét.

Ha az iteratív eljárást az alfa-béta algoritmussal kombináljuk, annak hátrányai is vannak. Az alfa-béta algoritmus ugyanis — ellentétben a minimax eljárással — csak a legjobb lépésnek ad korrekt pontértékeket. Azt tudjuk tehát, hogy mi a legjobb lépés az adott szintű kutatás során, de a többiről nem tudunk egyebet, csak hogy azok nem a legjobb lépések. Mindössze annyi információnk van róluk, hogy az adott pontérték, amit az alfa-béta keresés során kaptak, a maximális. Az biztos, hogy a lépés valódi értéke ennél nem nagyobb — ez a levágások következménye. Ha ugyanis az algoritmus — nem a legjobb gráf elemzése közben — egy elég jó lépést talált, akkor a fa többi részét levágja, hiszen annak elemzésére már nincs szükség. Ilyenkor ennek az „elég jó lépésnek" a pontértéke kerül egy szinttel feljebb, annak ellenére, hogy legtöbb esetben létezik ennél magasabb pontértékű lépés is.

Láthatjuk tehát, hogy a levágások miatt az alfa-béta algoritmus csak a legjobb lépésről ad felvilágosítást — kivéve az egy féllépéses kutatást, ahol az értékelés módszere természetszerűleg megegyezik a minimaxszal —, és emiatt az iteratív eljárásnál alkalmazott lépés-sorbarendezés az egy féllépésesnél mélyebb kutatás esetén feleslegessé válik, sőt általában rossz eredményt ad. Ezt

illusztrálja a *táblázatokban* látható pontfelvitel, amely az *ábrán* bemutatott hadállásban sötét lépését választja ki: minél kisebb a hexadecimális előjeles szám, az értékelőfüggvény szerint annál jobb az éppen lépésre következő fél pozíciója.

Az előbbi probléma megoldására KEMPELEN ATARI programom fejlesztése közben két ötletem támadt. Az egyik egy kézenfekvő, egyszerű eljárás. A program az első féllépésű értékelés után sorba rendezi a lépéseket, majd mélyebb kutatások esetén nem rendezi újból az összes lépést, hanem csak a legjobbakat helyezi át

Az egy féllépéses kutatás eredménye

Ff8xb4	001b
Hc6xb4	0023
Hg8 – f6	002b
Ff8 – d6	002e
Ff7 – e7	002f
Hg8 – e7	0030
Ba8 – b8	0031
Fc8 – d7	0033
Hg8 – h6	0034
a6 – a5	0034
h7 – h5	0034
g7 – g6	0035
Ba8 – a7	0035
h7 – h6	0035
Ke8 – d8	0036
Fc8 – b7	003a
Ke8 – d7	003b
Hc6 – d8	003d
Ke8 – e7	003d
Hc6 – b8	0041
Hc6 – a7	0042
Hc6 – e7	0042
f5 – f4	0049
g7 – g5	004b
Hc6 – d4	0069
Hc6 – e5	0073
Hc6 – a4	0074
Fc8 – e6	0076
Ff8 – c5	007a

A két féllépéses kutatás eredménye

Ff8xb4	0022	
	He2 – d4	ffde
	Hf3 – d4	ffe0
	He2 – c3	ffe0
	He2 – f4	ffe0
	Fb3 – d5	ffe1
	c2 – c3	ffe1
	Fb3 – c4	ffe2
	0 – 0	ffe3
	h2 – h4	ffe3
	Ke1 – d1	ffe4
	h2 – h3	ffe4
	g2 – g3	ffe5
	Bh1 – g1	ffe5
	Bh1 – f1	ffe5
	Ba1 – a4	ffe6
	Ba1 – a2	ffe6
Hg8 – h6	0022	
	Fb3 – d5	ffde
Hg8 – e7	0023	
	0 – 0	ffdd
Hc6xb4	0025	
	0 – 0	ffdb
Ff8 – d6	0029	
	Fb3xg8	ffd7
Ff8 – c5	0029	
	Fb3xg8	ffd7
Hc6 – d4	002b	
	Fb3 – a4 +	ffd5
Hc6 – a7	002b	
	Fb3 – a4 +	ffd5
Ba8 – b8	002c	
	Fb3xg8	ffd4
Ff8 – e7	002c	
	Fb3xg8	ffd4
Hg8 – f6	002d	
	c2 – c3	ffd3
Fc8 – d7	002e	
	Fb3xg8	ffd2
f5 – f4	002e	
	Fb3xg8	ffd2
a6 – a5	002f	
	Fb3xg8	ffd1
Hc6 – d8	002f	
	Fb3 – a4 +	ffd1
h7 – h5	002f	
	Fb3xg8	ffd1
g7 – g6	0030	
	Fb3xg8	ffd0
g7 – g5	0030	
	Fb3xg8	ffd0
Ba8 – a7	0030	
	Fb3xg8	ffd0
h7 – h6	0030	
	Fb3xg8	ffd0
Ke8 – d8	0031	
	Fb3xg8	ffcf
Hc6 – e7	0032	
	Fb3 – a4	ffce
Fd8 – b7	0035	
	Fb3xg8	ffcb
Hc6 – b8	0035	
	Fb3 – a4	ffcb
Ke8 – d7	0036	
	Fb3xg8	ffca
Hc6 – e5	0037	
	Fb3 – a4	ffc9
Ke8 – e7	0038	
	Fb3xg8	ffc8

az előző kutatás alapján már rendezett lépéslistán. Ennek egy továbbfejlesztett változata — amit tudomásom szerint más programban még nem alkalmaztak — abban különbözik az előbbi ötletemtől, hogy a legjobb lépés után következőknek is meghatározott korrekt pontértéket ad. Ezáltal ismét úgy rendezhető az összes lépés, hogy nemcsak a legjobbnak a helye változik meg a listán, hanem a többié is, az állásértékelésnek megfelelően. Ezt a majdnem korrekt pontértéket úgy kapom, hogy abban az esetben, ha az algoritmus egy vágó lépést talált, ennek pontértékével nem foglalkozik, hanem meghagyja az egy szinttel feljebb levő értékelés pontját. Abban az esetben viszont, ha egy pozícióban nincs vágó lépés, és emiatt az összes lépés értékelése megtörtént, a minimax elvnek megfelelően a legjobbnak talált lépés pontértékét egy szinttel feljebb visszük.

Ezt a módszert elég hatásosnak tartom: a játékfa megtartja teljes rendezettségét az elemzés végéig, továbbá lehetőséget ad arra is, hogy minden esetben összehasonlíthassuk a legjobb lépést követő pontértékeket, amiknek különbségeit felhasználhatjuk az értékelésnél és a végső lépés kiválasztásánál is (lásd Mikroszámítógép Magazin, 1985/2. szám).

Két féllépéses kutatás esetén egy kevésbé jó sötét lépésre egy közepes értékű világos válaszlépés is elég erős lehet ahhoz, hogy levágja a fa többi részét. Mivel világos nem az állásban megtehető legjobb lépésével vágta le a fát, ezért sötét számára kedvező pontértéket ad az állásértékelés. Ennek következménye, hogy két féllépéses kutatás után már nem a Hc6 × b4-et tekinti a program a második legjobb lépésnek, hanem a Hg8—h6-ot. E lépés megtétele után ugyanis a program az Fb3—d5 válaszlépést értékelte, a hadállásban megtehető valóban legjobb c2—c3 lépés helyett.

KOVÁCS P. ATTILA

Aprócska játékos trükkök, melyek szórakoztatnak, és egy picit a matematikai ismereteket is feleleventik

Petike kártyázik

Petike szeret kártyázni, de ritkán van partnere: apuka és anyuka soha nem ér rá, barátai pedig inkább számítógépes játékokkal szórakoznak. Így magára maradt hobbijával, a logikai feladatok megoldásával. Készített hát egy kártyázó programot. Próbálkozásához a „zsírozást" választotta. Ennek a játéknak a szabályait bizonyára sokan ismerik, de azért szólunk róluk.

A zsírozást a 32 lapos magyar kártyával játsszák. A játékban a színeknek nincs jelentőségük, csak a figuráknak: az a játékos győz, aki több „zsírt", azaz tízest és ászt gyűjt össze. Ezek a legértékesebb lapok tíz pontot érnek. Fontos szerepe van még a hetesnek is, amely nem ér ugyan pontot, de minden lapot üt. Egyébként ütni csak ugyanolyan lappal lehet, és ha a játékos bírja tartalékkal, akkor ismételhet.

Petike programja megszerkesztésénél a játék menetéből indult ki. Keverés után mindkét játékos négy-négy kártyát kap, a többit pedig lefelé fordítva az asztalra teszik. Az egyik játékos hív egy lapot. Ezt a partner vagy üti, vagy rátesz bármilyen másikat. Az utóbbi esetben a hívóé az ütés. Ha a másik játékos ütött, akkor az elsőnek joga van tovább ütni ugyanazzal a lappal vagy

```
10 REM ZSIROZAS
80 DIM A(4),B(4),K(32),K$(8),V(32),KK(32),L(8)
90 I=0
100 FOR I=1 TO 32
110 K=K+1:K(I)=K
120 IF X>7 THEN X=0
130 NEXT I
150 REM KEVERES
160 PRINT "TURELEM KEVEREK"
170 FOR I=1 TO 32
180 V(I)=0
190 NEXT I
200 RANDOMIZE
210 X=INT(100*RND)
220 IF X>32 THEN GOTO 210
230 FOR I=1 TO 32
240 IF V(I)=X THEN GOTO 210
250 NEXT I
260 FOR I=1 TO 32
270 IF V(I)<>0 THEN GOTO 290
280 V(I)=X
285 GOTO 210
290 NEXT I
300 FOR I=1 TO 32
310 H=V(I):KK(H)=K(I)
320 NEXT I
400 REM JATEK KEZDODIK
410 FOR I=1 TO 4
420 A(I)=0:B(I)=0
430 NEXT I
440 Z=1:AD=0:BD=0:D=0
445 BEEP
450 INPUT "AKARSZ KEZDENI? I/N",N$
460 IF N$="N" THEN GOTO 570
465 T=0:H=0
470 GOSUB 5800
475 D=0:GOSUB 5300
480 Y=1:GOSUB 5000
490 GOSUB 5900
500 Y=2:GOSUB 5000
510 INPUT "HIVJAL LAPOT",H
520 GOSUB 5500
525 GOSUB 5800
530 INPUT "TOVABB,HIVAS,VERE T/H/V",M$
540 IF M$="H" THEN AD=AD+D:GOTO 470
550 IF M$="T" THEN GOTO 460
560 BD=BD+D
570 D=0:GOSUB 5300
580 Y=1:GOSUB 5000
590 GOSUB 5900
600 Y=2:GOSUB 5000
610 REM GEP KEZD
620 N1=0:N4=0:N8=0
630 FOR I=1 TO 4
640 IF B(I)=1 THEN N1=N1+1:GOTO 670
650 IF B(I)=4 THEN N4=N4+1:GOTO 670
660 IF B(I)=8 THEN N8=N8+1
670 NEXT I
680 M4=N1+N4:M8=N1+N8
685 IF M4>=3 THEN P=4:GOTO 700
690 IF M8>=3 THEN P=8:GOTO 700
695 GOTO 720
700 FOR I=1 TO 4
705 IF B(I) <> P THEN GOTO 715
710 T=I:PRINT"LAPOM   ",K$(I)
711 GOTO 900
715 NEXT I
716 REM VALAMIT HIVNI KELL
720 FOR I=1 TO 4
730 IF B(I)=1 THEN GOTO 785
740 IF B(I)=4 THEN GOTO 785
750 IF B(I)=8 THEN GOTO 785
770 T=1:PRINT "LAPOM   ",K$(I)
780 GOTO 900
785 NEXT I
790 FOR I=1 TO 4
800 IF B(I)=1 THEN GOTO 840
820 T=I:PRINT "LAPOM   ",K$(I)
830 GOTO 900
840 NEXT I
900 INPUT " MI A VALASZOD?",H
905 IF H=0 THEN GOTO 1100
910 IF A(H)=B(I) THEN GOTO 960
920 IF A(H)=1 THEN GOTO 960
930 GOSUB 5800
940 BD=BD+D:GOTO 570
960 IF B(T)=4 THEN GOTO 1000
970 IF B(T)=8 THEN GOTO 1000
980 A(H)=0:B(T)=0
990 PRINT "NEM TESZEK LAPOT"
995 GOTO 470
1000 GOSUB 5800
1010 FOR I=1 TO 4
1020 IF B(I)=1 THEN T=I:GOTO 1060
1030 NEXT I
1040 PRINT "NEM TESZEK LAPOT"
1045 GOSUB 5800
1050 AD=AD+1:GOTO 470
1060 PRINT "LAPOM   ",K$(T)
1070 GOTO 900
1100 IF Z<> THEN GOTO 560
1110 IF AD<BD THEN GOTO 1140
1120 PRINT "GRATULALOK GYOZTEL"
1125 PRINT "PONTJAID",AD
1130 GOTO 150
1140 PRINT "SAJNALOM EN GYOZTEM"
1145 PRINT "PONTJAIM   ",BD
1150 GOTO 150
5000 REM AZONOSITAS
5050 IF Y=2 THEN GOTO 5100
5060 FOR I=1 TO 4
5070 L(I)=A(I)
5080 NEXT I
5090 GOTO 5130
5100 FOR I=1 TO 4
5110 L(I)=B(I)
5120 NEXT I
5130 FOR I=1 TO 4
5140 IF L(I)=0 THEN K$(I)="URES":GOTO 5280
5150 IF L(I)=1 THEN K$(I)="VII":GOTO 5280
5160 IF L(I)=2 THEN K$(I)="VIII":GOTO 5280
5170 IF L(I)=3 THEN K$(I)="IX" :GOTO 5280
5180 IF L(I)=4 THEN K$(I)="X" :GOTO 5280
5190 IF L(I)=5 THEN K$(I)="ALSO":GOTO 5280
5200 IF L(I)=6 THEN K$(I)="FELSO":GOTO 5280
5210 IF L(I)=7 THEN K$(I)="KIRALY":GOTO 5280
5220 K$(I)="ASZ"
5280 NEXT I
5290 RETURN
5300 REM LAPOK HUZASA
5301 IF Z=32 THEN GOTO 5350
5305 FOR I=1 TO 4
5310 IF A(I)=0 THEN A(I)=KK(Z):Z=Z+1
5320 IF B(I)=0 THEN B(I)=KK(Z):Z=Z+1
5330 IF Z=32 THEN GOTO 5350
5340 NEXT I
5350 RETURN
5500 REM A GEP VALASZA
5501 FOR I=1 TO 4
5505 IF A(H)<>B(I) THEN GOTO 5520
5510 T=I
5515 PRINT "VALASZOM",K$(I):GOTO 5750
5520 NEXT I
5525 REM ELSO LAP-E
5530 FOR I=1 TO 4
5540 IF B(I)=0 THEN GOTO 5570
5550 NEXT I
5560 GOTO 5650
5570 IF A(H)=4 THEN GOTO 5590
5580 IF A(H)=8 THEN GOTO 5590
5585 GOTO 5650
5586 REM VAN-E VII-ES
5590 FOR I=1 TO 4
5600 IF B(I)=0 THEN GOTO 5640
5610 IF B(I)<>1 THEN GOTO 5640
5620 T=I:PRINT "VALASZOM",K$(I)
5630 GOTO 5750
5640 NEXT I
5641 REM BARMILYET
5650 FOR I=1 TO 4
5660 IF B(I)=0 THEN GOTO 5720
5670 IF B(I)=1 THEN GOTO 5720
5680 IF B(I)=4 THEN GOTO 5720
5690 IF B(I)=8 THEN GOTO 5720
5700 T=I
5710 PRINT "VALASZOM",K$(I):GOTO 470
5720 NEXT I
5725 FOR I=1 TO 4
5730 IF B(I)=0 THEN GOTO 5740
5731 T=I
5735 PRINT "VALASZOM",K$(I):GOTO 470
5740 NEXT I
5750 RETURN
5800 REM PONTOK SZAMOLASA
5810 IF A(H)=4 THEN D=D+10
5820 IF A(H)=8 THEN D=D+10
5830 IF B(T)=4 THEN D=D+10
5840 IF B(T)=8 THEN D=D+10
5850 A(H)=0:B(T)=0
5860 RETURN
5900 REM LAPOK KIJELZESE
5910 FOR I=1 TO 4
5920 PRINT
5930 PRINT I,"SZAMU LAP",K$(I)
5940 NEXT I
5950 RETURN
```

hetessel. A játék egy-egy fordulójában legfeljebb négy ütés lehet, mert ennyi a kézben levő lap. Ha valamelyik játékos nem üt tovább, akkor kiegészítik a kézben levő kártyák számát négyre. Ha az utolsó menetben a lefordított lapok száma kevesebb, mint kellene, akkor annyit vesznek fel belőle, hogy lapjaik száma egyenlő legyen. A játék végén megszámolják „zsírjaikat".

Érdemes néhány partit kártyával lejátszani, hogy a program logikája érthetőbbé váljon.

Petikének először a keverést kellett megoldania. Ezt végzik el a 170—320-as sorok: a K tömb elemeit véletlenszerűen feltöltik 1—32-ig terjedő számokkal, és elemeihez rendelik a magyar kártya lapjait. A „keverés" viszonylag hosszú időt igényel, amely csökkenthető, ha gépünk véletlenszám-generátorát az 1—32 értékhatár között állítani tudjuk.

Petike a gép „gondolkodásának" programozásánál az élő játékos logikájából indult ki. Amikor a játékos hív, a gép az 5500-as programsortól kezdődően a következőket teheti. Ha van azonos lapja, akkor üt, és a játékosé a döntés joga. Ha nincs, megvizsgálja, hogy értékes-e a hívott lap. Ha igen, megpróbál hetessel ütni (5570—5640-es sor). Egyéb esetben nincs mit tenni, lerak valamilyen lapot, de lehetőleg nem „zsírt". Továbbütésnél ez az eljárás ismétlődik.

Híváskor a számítógép úgy „gondolkodik", hogy csak akkor célszerű értékes lappal indulni, ha abból a játékosnak hetesekkel együtt legalább három van a kezében, mert a partnernek is lehet belőle; egyébként értéktelen lappal kell kezdeni (620—710-es sorok). Az ütés után az 5300-as sorral kezdődő szubrutin kiegészíti a játékosok lapjait négyre, az 5800-as sor pedig megszámolja pontjaikat. Az 5000-es sorral kezdődő szubrutin a partnerek lapjait azonosítja a dobások és húzások után.

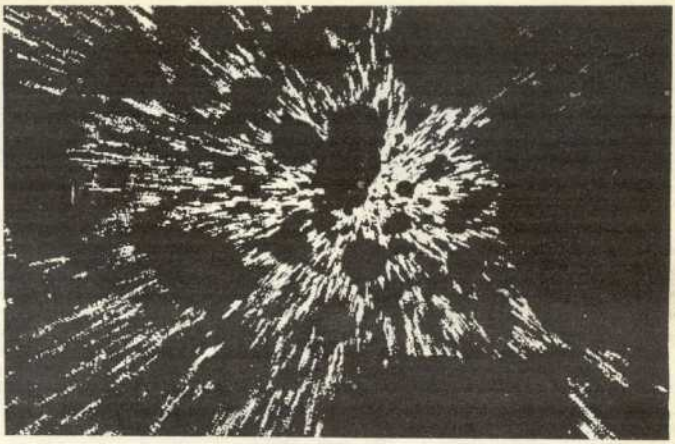
A program MS BASICA nyelven készült, a lehetőség szerint egyszerű utasításokkal. A játék hardver megvalósítása megtalálható Kovács Mihály Néhány kibernetikai játékgép című könyvében (Budapest, 1969. Tankönyvkiadó).

Az alkalmazott jelölések a következők. A K tömb a magyar kártya kódolt lapjait tartalmazza: VII=1, VIII=2, ... ász=8. A KK tömbben a már megkevert pakli, az A és B tömbben a játékosok kezében lévő kártyák vannak.

A program szubrutinokra épül. Ez a rövidítésen kívül lehetővé teszi a játékszabályok változtatását is: például bővíthető a játék azzal az engedménnyel, hogy nem kötelező azonos lapot tenni, ha van ilyen. A program keverést végző része más kártyajátéknál is felhasználható.

Szívesen fogadunk ötleteket és programokat egyéb kártyajátékokról.

PINKE GYÖRGY



A természet vigyáz ránk

— Sietned kell, hallod! Igyekezz!
2100. 11. 16
Híreket mondunk.

A legutóbbi felmérések szerint a világ népessége elérte a 8 milliárdot. A Biológiai Kutató Intézet közlése szerint elenyésző, 0,1 százalék azoknak a születéseknek a száma, melyek során csak egyetlen gyerek születik. Ezzel szemben 50,7 százalék a kettes, 30,8 százalék a hármas ikrek születési aránya. A fennmaradó szüléseknél háromnál több gyermek látja meg a napvilágot.

Érdekes eseményekről számol be a UPI hírügynökség. Közlése szerint a Sziklás-hegységben felfedeztek egy törzset, tagjai különös képességgel rendelkeznek: talpuk felé tudják fordítani a lábujjaikat, kapaszkodnak velük.

— Gyerünk, lassú vagy!
2105. 11. 16.
Híreket mondunk.

A legutóbbi felmérések szerint a világ népessége 10,5 milliárdra tehető. A Biológiai Kutató Intézet felmérése kimutatta, hogy már a nyolcéves gyermekek nemzőképesek.

A TASZSZ közlése szerint olyan új vércsoportba tartozó emberekre bukkantak, akiknek szervezete ellenáll a sugárfertőzésnek.

— Megint elkésel, siess!
2120. 11. 16.
Híreket mondunk.

A legújabb kutatások szerint a világ népessége eléri a 20 milliárdot.

A két szemben álló ország között a helyzet egyre feszültebb.

Budapest. Az MTI idegen forrásra utalva közli, hogy egy Csendes-óceánban fekvő szigeten olyan embercsoportot találtak, amelynél kettős légzés alakult ki. Kopoltyúval és tüdővel is tudnak lélegezni.

A Biológiai Kutató Intézet felmérése szerint az embereknek egyre javul a szaglásuk, de ezzel párhuzamosan romlik a látásuk.

— Siess!
— Ne dumálj bele, mindig miattad rontom el!
2200. 11. 16.
Híreket mondunk.

Készültségbe helyezte nukleáris elhárító rendszerét az egyik ország. A közvélemény-kutató irodák szerint az atomháború kitörésének lehetősége 97 százalék.

A Biológiai Kutató Intézet szerint az emberi tűrőképesség határa jelentősen kitolódott. Képesek vagyunk például, három órán keresztül 70 Celsius-fokot elviselni, és négyheti szomjazást.

— Megint buksz!
— Kuss!
2225. 11. 16.

Híreink előtt hallgassák meg szerkesztőnk jegyzetét, A természet vigyáz ránk — címmel.

A természet vigyáz ránk. Igyekszik megóvni bennünket az atomháború borzalmaitól. Ellenállóvá teszi a növényeket, az állatokat és ...

Elnézést, de meg kell szakítanunk adásunkat.
Fontos közleményt kell beolvasnunk.
Kitört a háború!
A képernyőn felirat jelenik meg.
Ketten figyelték.

Az atomháború lefolyt.
Túlélők száma=0
Vesztett!
Akar még egyet játszani? I/N

HARANGOZÓ ANDRÁS

Már a µM előző havi számában is közöltünk egy levelet, amely a programok illegális másolásával és terjesztésével foglalkozik. Az egyik szerkesztőségi írásom (Copyright — µM 1985/2. szám) is elég nagy visszhangot váltott ki, a cikk megjelenése után meglehetősen sok telefont kaptam. A telefonálók egy részének a stílusa elég lekezelő volt („mit ugrál, ez ma a legnagyobb problémája a hazai számítástechnikának?"), de sokan hasonlóan gondolkodtak, mint a már megjelent és a most következő levél írója. Ha olvasóink rászánnak egy levél megírására való időt, szívesen közölnénk a problémával (vagy az álproblémával?) kapcsolatos véleményüket.

Imhof Iván, Dombóvár,

Móricz Zs. u. 11. 7200

Egy komoly problémával kapcsolatban szeretném a véleményét és tanácsát kérni. Azt hiszem — bár ez esetben személyes jellegű — a számítástechnikát kedvelők és művelők népes táborát is érinti. Konkrétan a programok „illegális" másolásáról és a szoftvervédelemről van szó.

Jelenleg főiskolás vagyok, előtte egy évig operátorként dolgoztam, így az élet több területéről rendelkezem némi tapasztalattal a témát illetően. Azt hiszem köztudott, hogy az amatőr — hobby és tanulási célból — programozók körében a programok másolása és cseréje elfogadott és természetes dolog, bár jogilag talán vitatható. Még inkább az, ha a jelenséggel a profi programozóknál találkozunk, ami (szerintem) szintén nem ritka. Nálunk a főiskolán a programok úgy terjednek, mint az influenza. C64 gépre szinte az összes komoly program könnyedén megszerezhető (GEOS, Art Studio, Newsroom stb.). Azt hiszem, az említett körökben senkinek nem okoz lelkiismereti válságot a tudat, hogy kalózmásolat van a birtokában. Eddig magam is így voltam ezzel, bár tudtam, hogy a szoftver éppen úgy szellemi termék, mint a könyv, vagy egy zenemű.

Ennyit bevezetésül, a most következő rövid történetben pedig szándékosan nem akarok neveket említeni.

Nemrég másoltam egy rendszerprogramot, mellyel nyári gyakorlatomon szerettem volna dolgozni. Leírásom nem volt hozzá, de emlékeztem, hogy egy ismert szoftverforgalmazó cég hirdette a Mikromagazinban. Meg is találtam a hirdetést, majd felkerestem az illető céget. Elmondtam, hogy egy bizonyos programhoz leírást szeretnék venni. Teljesen jóhiszeműen úgy gondoltam, hogy ha a könyvesboltokban számtalan felhasználói programról kapható leírás, ez egy ilyen cégnél is lehetséges. A céget képviselő úr megkérdezte, miért nincs leírás, hiszen ők minden eladott programhoz adnak. Megmondtam, hogy nem tőlük vettem, másolással került hozzám. Erre az úr felháborodva, de udvarias hangnemben közölte, hogy szó sem lehet róla, ő nem segédkezhet egy szoftverlopásban, és egyáltalán hogyan képzeltem, hogy a forgalmazó céghez jövök a lopott szoftverhez leírást kérni. Én kövér dermedve álltam, hiszen bennem ez még sohasem vetődött fel ilyen komolyan. Kínos magyarázkodásba kezdtem: tudom, hogy van szoftver jogvédelem, és jogi-erkölcsi szempontból helytelen a másolás, de ha már megvan, szeretnék dolgozni vele. Az úr megjegyezte, hogy ez teljes képtelenség, és neki tulajdonképpen jegyzőkönyvet kellene felvenni, miképp jutottam én illegálisan egy ilyen programhoz. (Csak mellékesen mondom el, hogy vita közben elismerte: néhány programhoz ő sem pénzért jutott hozzá. Ez számomra nem lenne szilárd erkölcsi alap a hasonló eljárások kritizálására.) Rövid vita után, melyben szavai nyomán szinte veszélyes bűnözőnek éreztem magam, távoztam.

Ennyi. Természetesen nem tudtam gyorsan naprendre térni a dolog felett, megpróbáltam több szempontból is végiggondolni.

A szoftvervédelem jogi szabályzásával nem vagyok tisztában — gondolom, sok amatőr programozóhoz hasonlóan. Így nem tudom, mit gondoljak: tényleg jogilag büntetendő lopást követtem el, mikor lemásoltam egy programot? Az számomra is világos, ha valamely nem kereskedelmi úton szerzett szoftver segítségével netán én készítek egy eladható programot, az ebből származó hasznom végül is az illető szoftverkészítőt károsítaná. Sőt, már az is, hogy nem pénzért jutottam hozzá. Valószínűleg én is így érezném, ha egy általam készített, kereskedelmileg forgalmazott program (sajnos, ilyen még nincs) kalózmásolatával találkoznék.

Ez az egyik oldal. A másik: a szoftverek másolás elleni védelme technikailag meglehetősen nehéz, mivel majdnem minden kód megfejthető; ha azonban mégis sikerülne, annak kellemetlen következményei is lennének. A nem zseniális programozók nem jutnának hozzá a nélkülözhetetlen segédprogramokhoz (fordítók, belövők stb.), az egyéni felhasználók pedig a felhasználói programokhoz (szövegszerkesztők, adatkezelők stb.), hiszen az árak a többezer Ft-os nagyságrendnél kezdődnek. Én sem írhatnám levelem Easy Script-tel (ha jól tudom, kb. 10 000 Ft lehet). Szerintem a számítástechnika fejlődésében a fiatalok — elsősorban a diákok — rendkívül nagy szerepet játszanak új iránti fogékonyságuk, ötleteik miatt, ezzel pedig pont ezt a réteget fosztanák meg sok lehetőségtől.

Visszatérve a velem történtekre: ha a cég a másolást megakadályozni nem tudja, az lenne a helyes út, hogy „zárolja" a rá vonatkozó információkat? Ezzel lassítja ugyan a programmal való munkát, de megakadályozni nem tudja, mert akit igazán érdekel, az nem riad vissza, és előbbutóbb magától kiismeri a rendszert. Így viszont csökken a programírásra fordítható idő. Számomra ez kissé a cég „bosszújának" tűnik, és nem értem, miért jobb így nekik. Talán ez is jogi kérdés lenne? Úgy vettem észre, könyvesboltjainkban a program megvásárlása nélkül is vehetek számtalan programleírást (Simon's, Help+, Easy, Master stb.).

A Mikromagazinban is volt egy rövid ismertető erről a programról. A cikk végén „A téma iránt érdeklődőknek a következő szakirodalmat ajánljuk" címszónál több külföldi és magyar szakkönyv között megtalálható volt a fejlesztő és a forgalmazó cég által kiadott kézikönyv is. Ezek szerint az érdeklődőnek a programot is meg kell vennie, ha részletes információhoz akar jutni róla?

Összefoglalva: mennyiben törvényellenes a szoftver ilyen formájú másolása? Lehetne-e olyan megoldást találni, ami sem a szoftverkészítők, sem a kispénzű felhasználók érdekeit nem sérti? Van-e kilátás rá, hogy a jövőben programot, szakirodalmat — esetleg még számítógépet is — nem a nyugati ár három-négyszereséért lehessen megvenni?

Most az egyszer itt nem válaszolok a levelére, mert azt szeretném, ha olvasóink küldenék el véleményüket és válaszaikat.

Mocsáry Gábor, Budapest,

Tűzoltó u. 8. 1094

Azt hiszem, hogy találtam egy tartósított „NOVOTRADE gumicsont"-ot. Ebből rengeteg van, bárki meggyőződhet róla, ha megveszi a DATA BECKER—NOVOTRADE fordításokat. Azért nem nevezem könyveknek, mert a könyveket általában olvasni és érteni lehet, a fordítások meg olyanok, amilyenek.

A történetem röviden a következő. A COMMODORE 64 létezésétől fogva keresem hozzá az irodalmat, így az említett nyomdaipari termékeket is megvettem. Ezeknél jobb programvicceket és fejtőröket még kitalálni sem lehet. Kezdetben nagyon bosszantott a dolog, de később rájöttem — és kedvenc időtöltésemmé vált —, hogy a hibák keresgélése közben lehet a legjobban tanulni. Most azonban megakadtam.

A „Tippek és trükkök a C64-hez" 55. oldalán és a „A VC—1541-es lemezegység programozása" 255. oldalán van két azonos elvű, a MID$-et átalakító gépi kódú program. A jól beépített hibák miatt gyakorlatilag egyik sem működik, sőt mindkettő alkalmas arra, hogy a gépnek azonnali „agygörcsöt" okozzon. A hibákat nem sorolom fel, mert túl hosszú volna. Mivel a programok alapötletét jónak találtam, nekiálltam, hogy „beinditom" őket. Ekkor értek a meglepetések.

A javítások után a fordítást elsőnek a HELP PLUS-szal, másodiknak a PROFI—ASS 64-gyel próbáltam. Amíg maga a lista hibás volt, addig a fordítókkal nem volt baj, csak a programok nem működtek. Ahogy az utasítások és a címek a helyükre kerültek, a két fordító megbolondult. Például a mellékelt, a HELP PLUS-szal készített listán a 240-edik sorig nincs baj. Ettől kezdve a fordító a 240-es sor BEQ, a 350-es sor BEQ, a 400-as sor BNE, a 420-as sor BNE, az 530-as sor BCS, a 680-as sor BEQ, a 700-as sor BCS és a 830-as sor BNE utasításainál — mindegyik összehasonlító! — a címeket rendre rosszul számította ki. A listán a címek jó helyen vannak, az ugrási címekre gondolok, az elvileg hibátlan listákon viszont már nem. A jelenség azért is furcsa, mert a PROFI—ASS 64 listája is teljesen azonos bolondéria szerint alakul.

Mivel a fordítók nem működtek jól, elkészítettem a BASIC betöltőt, a helyes ugrási címekkel.

A program így betöltve hibátlanul működik, amit a mellékelt REC/2-TEST programmal ellenőriztem.

A tudományom itt megállt! Sajnos nem jöttem rá, mi okozhatja a rejtélyes eseményeket. Ha valaki megfejtené a talányt, annak nagyon örülnék. Bizonyára sokan dühöngenek a NOVOTRADE program viccei miatt. Azt már meg sem merem kérdezni, hogy mivel készítik a hitelesnek eladott listákat (lásd a Tippek és trükkök 57. oldal 860-as programsorát).

Kérésem az, hogy ha a témával nem foglalkoznak, erről egy rövid levélben értesítsenek. Amennyiben a megoldás érdekelné Önöket, és további adatokra volna szükség, úgy állok rendelkezésükre.

Levelét elküldtem a Novotrade igazgatójának, aki mindig gondoskodik arról, hogy a levélre munkatársai válaszoljanak. Íme a könyv lektorának válasza: Sajnos ki kell ábrándítanom, mert a „tartósított NOVOTRADE gumicsontot" nem sikerült megtalálnia. Ellenben nekem okozott egy kis fejtörést az a lista, amelyet levelében mellékelt. A program ön szerint a „Tippek és trükkök a C64-hez" c. könyvből való. Nos némi eltérés tapasztalható a mellékelt és a könyvbeli program között. Valószínűleg figyelmetlenségből ered, hogy programjában a 150. sor (@35A címen): JSR $B08D utasítás szerepel, míg ha megnézi, a könyvben ez az utasítás: JSR $B08B! Nos ez az egyik oka annak, hogy programja nem működik. Sokat szoktam én is bosszankodni a figyelmetlen begépelések okozta

„programhibák" miatt, ezek nem „programviccek", hanem emberi tévedések. Annak érdekében, hogy számítógépe ne kapjon „azonnali agygörcsöt", a következőket tudom tanácsolni: az utasítások begépelésénél kövesse a könyvben megadott sorrendet. Tehát a programjában a 85Ø. sortól (= $Ø3 . . . stb.) kezdődő utasításokat helyezze a program elejére, mint ahogy azt a szerzők is tették, így a HELP PLUS is jól megbírkózik a címek helyes kiszámításával. Ezen a problémán nem kell „dühöngeni", „bosszankodni", ez a HELP PLUS ill. a PROFI—ASS 64 „hibája".*

Pataki János, Kiskunfélegyháza,

Horváth Z. út 2. 6100

A spectrumosok széles táborához tartozom, és gondjaim hasonlóak a többiekéhez. Én nem keveslem a Spectrum cikkeket, inkább azzal vagyok elégedetlen, amiről szólnak. Véleményem szerint programokat fölösleges közölni két okból is. Egyrészt egy valamire való program terjedelme miatt nemigen való újság hasábjaira, másrészt nem valószínű, hogy valaki begépeli. Ezenkívül bármilyen problémára több száz profi program forog közkézen. A baj az, hogy megfelelő leírás hiányában nagy részüket lehetetlen használni. Ezt kellene a lapnak fölvállalnia, és közreadni programok leírásait. Tudom, hogy a Spectrum Játék és Program című könyv 3 kötete ezt a feladatot vállalta, de mivel ismertek a könyv megjelenési idők, az átfutás nagyon hosszú, sok program elveszti aktualitását. Főleg felhasználói programok leírására gondolok, pl. ANIMATOR 1, GRAPHICS ADVENTURE CREATOR, WHITE LIGHTNING, HP—80, BETA BASIC 3.2, ARTIST—II., de lehetne a sort nagyon sokáig folytatni. Ezenkívül sokan örülnének különböző játéktérképeknek, akár valamilyen külföldi lapból kölcsönvett, akár az olvasók által készített. Továbbá közzé lehetne tenni a legújabb örökélet POKE-okat.

Javaslom továbbá, hogy a két legelterjedtebb gép, a C64 és Spectrum legújabb programjairól legyen TOP lista. Gondolom ez igen egyszerűen megvalósítható — és szinte mindenki szívesen látná — hiszen szinte minden külföldi számítógépes lap közöl ilyeneket. Ez már csak azért is célszerű lenne, mert innen lehetne informálódni a legújabb programok megjelenéséről.

Másnak más a véleménye, nagyon sok olvasónk megírta, hogy bizony begépelik az általunk közölt programokat. Ezt persze abból is észleljük, hogy nagyon sokan elverik a port rajtunk, ha a közölt programokban hiba van. A többi javaslatán elgondolkozunk.

Ifj. Szabó József, Oroszlány,

Harasztthegyi út 1/C 2840

Hadd kezdjem levelemet azzal, hogy ZX81-em hősi halált halt. Mégpedig úgy, hogy a modulátort ki kellett szedni azért, mert csak így tudtam megmenteni a ZX SPECTRUM-omat, mert a speci-é elromlott, és csak drága áron kaptam volna egy új modulátort. Szóval a speci — már működik (egy élet árán!), csak a tv-zsinór szakadt el, és javíthatatlan lett. Kérem Önt, hogy szóljon a Sinclair Service-nek, hogy küldjenek nekem 200 Ft ellenértékben, utánvétellel egy gyári kábelt. Speci-re. Nagyon megköszönném, ha szólna nekik. Vagy valamelyik olvasóval lerendezném, hogy ha nem tudná Ön lerendezni.

Vennék Mega Basic 4.0 és Beta Basic 1.8 programokra könyvet. Mert könyv nélkül nem tudom használni ezeket a programokat. Ugyanakkor levelezném program- és tapasztalatcsere céljából is.

Szívesen közlöm a levelét, de megrendelések elintézését nem vállaljuk, túl sok a dolgunk az újságszerkesztéssel.

Avarosi László, Budapest,

Hámán Kató u. 4. 1096

Megköszönöm a levelemre adott választ (Mikroszámítógép Magazin 1987/6.).

Úgy látszik, jócskán illúziókat tápláltam, amikor feltételeztem, hogy sokak rendelkeznek nyomtatóval. Bocsássa meg nekem, de én a folyóirat írásai mögött sejthető színvonalat általánosítottam. Ezért tápláltam magamban egyfajta irigységet, úgy éreztem, hogy „mindenkinek" van nyomtatója, csak nekem nincs. Ez a hitem most megdőlt.

Korábban a Z80-as mikroprocesszorral kívántam foglalkozni, ezért vásároltam a ZX Spectrumot. Időközben beláttam, hogy roppant előnyös volna hozzá egy nyomtató. Lélekben beletörődtem még az árakba is. Elvégre olcsóbb hobbi, mint a kertészkedés vagy autózás. Azonban a vásárlás megoldhatatlan idehaza, mindenféle zugkereskedésben feltűnnek ugyan gyanús portékák, melyekről többnyire még műszaki információt sem képesek mondani, nemhogy interface-t kínálni. Gondoltam, hogy valamely nyugatra utazó ismerősömet megkérem a vásárlásra, de kiderült, hogy a devizaszámlámról nem adhatok át összeget másnak. Jómagam közlekedési iszonyból eredően nem utazok.

A nehézségek ellenére remélem, hogy előbbutóbb lesz nyomtatóm, és akkor az Ön válaszában említett segítséget nyújthatom. A beszerzést illetően egy jó tippet örömmel fogadok.

Talán nem haragszik, ha megragadom az alkalmat két apróság visszajelzésére. A '87/6. szám 6. oldalán tapasztaltam eltéréseket a programlistában. Szerintem a 80-as DATA sorban levő 133-as szám helyesen 33, és a 89-es DATA sorban 60 helyett 80-as szám kell. A lista 10-es és a 40-es sorából a LET értékadó utasítás hiányzik.

A másik észrevételem a '87/7. szám 31. oldalán látható ZX81 sortörlő programra vonatkozik. Ez a frappáns kis segédlet jól alkalmazható a ZX Spectrum gépen is, hiszen a Basic utasítássor formátuma e két géptípusnál azonos, csupán a memóriacím kezdőértéke tér el. Tehát Spectrum esetén, a lista 6-os sorában C-nek 23 755 értéket kell adni (microdrive alkalmazásakor megint más értéket), s máris jól használható.

Végül hadd lapozom át ismét a MM legutóbbi '87/7. számát, megmutatva, mi érdekeset találtam.

2. old. — Szerintem a távbeszélő-hálózat hiánya még súlyos károkat és feszültségeket fog okozni az informatika terén. Nyilvános adatbázisok terén a lemaradás már elképesztő.

21 old. — görögországi beszámoló, tanulságokkal

23. old. — gondolatébresztő ez a sorozat a mesterséges értelemről

27. old. — az „implementáció"-t nem sikerült megfejtenem, szótárban sem találtam

36. old. — megkeresem a Video Elektronika gmk-t, hátha tudnak segíteni nyomtató ügyében

38. old. — végre megértettem a lapkezelési technikát

41. old. — Nagyszerű! Szókimondó, igaz. A hazai számítógépes kultúráról

43. old. — élvezetes beszámoló az „Atari Kempelen" küzdelméről.

Ezzel a kis szemlével igazolom, hogy a magazin ismét tartalmas volt.

Egyetértünk, hogy a hazai nyomtató-vásárlási lehetőségek ma egyre több problémát jelentenek nemcsak az amatőr mozgalomban, de pl. az iskolai számítógépek használatában is. Egy kevesebb, mint tízezer forintos gép mellé egy többször tízezer forintos nyomtató megvásárlása meggondolandó. Ugyanez a helyzet persze a floppy-val is, ennek a két perifériának a hiánya — véleményem szerint — egyre jobban akadályozza a házi számítógépek valóban hasznos alkalmazását (pl. könyvszerkesztés, adattárolás stb.). A megjegyzéseit és különösen a 7. számról írt szemléjét a szerkesztőség valamennyi munkatársa nevében köszönöm.

Nagy Zoltán, Budapest,

Árpád út 103. III. 3. 1042

Régi olvasója vagyok kitűnő lapjuknak annak ellenére, hogy talán én vagyok az egyedüli magyar, aki nem ért a számítástechnikához.

Viszont szeretném, ha gyermekeim komolyabb kapcsolatba kerülhetnének ezzel a szép tudománnyal. Mint „határidő-nélküli előjegyző" hiába vártam C 16-os vagy C plus 4-es gépekre, — a forgalmazó még azt sem árulta el, hányadik vagyok hasonló sorsú társaim között. Ezért örömmel fogadtam a Centrum Áruházak akcióját, és vettem egy ENTERPRISE 128 típusú gépet. Azóta nem tudom, helyesen cselekedtem-e, mert a hozzáértők sajnálkoznak és becsmérlően nyilatkoznak a gépről.

Azt hiszem többedmagammal szeretnénk olyan objektív értékelést megismerni, mely helyére teszi az Ep 128-at. A használatnál is felléphetnek olyan problémák, melyet nem tudunk megoldani. Ilyen gond például az, hogy nem tudjuk a színes tv-video üzem előnyeit hasznosítani, mert a tv-video csatlakozójára bevitt számítógép jel nem ad színes képet.

A szaksajtó ezzel a géppel egyáltalán nem foglalkozik. Kérem ezért tisztelt Főszerkesztő Urat, hogy nagyrabecsült lapjukban — pár sor erejéig — szíveskedjék foglalkozni az Ep 128 géppel, hiszen a hazai piacon ez az egyedüli, viszonylag hozzáférhető áron kapható számítástechnikai eszköz.

1. Nem Ön az egyedüli.

2. A μM 87/9. számában jelent meg a μ Klub rovatban a kért ismertető „Szinkronban az idővel" címmel.

3. Ami az Ep-t illeti, néhány barátom a gépet nagyon dicséri, különösen a szövegszerkesztővel elégedettek.

4. Sok olvasónk kérdezte, hogy mikor lesz az Ep kompatibilis a Sinclair Spectrummal, mert szeretnék használni a Spectrumhoz kapható programokat. Egyelőre csak annyit hallottunk, hogy: „majd".

*

Egy korábbi olvasói levelünkre a Novotradetől az alábbi választ kaptuk

Tisztelt Kovács Győző elvtárs!

Hivatkozva 1987. június 9-én kelt levelére, melyben Fischer Kornél — Commodore problémára kért tanácsának — levelét mellékelten megküldték, írásban a levélírónak — cím hiányában — nem tudunk válaszolni.

A kért kapcsolási rajzok Szervizünkben megtalálhatóak (Budapest, 1053 Magyar u. 1.)

A levélíró — tanácsoljuk — közvetlenül keresse meg a Szerviz vezetőjét (Hegedüs Mihályt), akinek saját hatáskörébe tartozik a szerviz-dokumentációk kiadása.

Üdvözlettel:

P. H.

Záng Zoltán számítástechn. üzletágvezető

Friedrichné Újvári Zsuzsanna számítástechn. üzletágvezetőh.

Köszönöm olvasóink változatlanul aktív és szíves érdeklődését

KOVÁCS GYŐZŐ

Tájékoztatásul közöljük, hogy — korábbi hirdetésünktől eltérően — a C Plus 4 (1987. szeptember 14-én) a Skála Sprint áruházban kapható. Ára 15 000 forint, a hozzá tartozó magnó 3500 forint.

Csikós Zsolt:
Zene C–64/128 kedvelőknek
(Budapest, 1987.
LSI ATSZ,
136 oldal. Ára: 125,— Ft.)

Sokak szerint a számítógéppel nem lehet igazán zenélni, nem lehet a hangszereket tökéletesen utánozni. A zene számokkal való leírása embertelenné teszi a hangzást — mondják —, és ebben van is némi igazság.

A számítógéppel való zenélés nem pótolhatja az élő zenét, de célja nem is ez, hanem új, szokatlan hanghatások keresése, a zenei lehetőségek szélesítése.

A szerző könyvének összeállításakor a BASIC nyelv ismeretét tételezte fel, remélve, hogy a programban előforduló trükkök a jártasabb olvasók számára is hasznosak lesznek.

A kötet első fejezete az olvasót a zeneelméleti alapismeretekbe és érdekességekbe vezeti be, majd ismerteti a SID 6581 chipet. Számos példát mutat be dallamok programozására, ismerteti a szintetizátort. Külön fejezet foglalkozik a zene és a grafika kapcsolatával.

Szűcs Pál:
Személyi számítógépek az oktatásban (Budapest, 1986. OMIKK,
242 oldal. Ára: 98,— Ft.)

A 80-as évek elején a számítógépek megjelentek az oktatásban is. A számítógépek oktatásbeli alkalmazásának legfontosabb vonása az, hogy a tanítási-tanulási folyamatban új lehetőségek, módszerek megjelenését teszik lehetővé, miközben az egyes tanulók igényeihez alkalmazkodnak. Az elterjedésük és gyakorlati felhasználásuk ütemét jelentősen gátolja az a tény, hogy világszerte, sőt még az egyes országokon belül is, különböző, egymással nem kompatibilis berendezéseket használnak. Jelenleg nincsenek érvényben megfelelő szabványok és ajánlások a hardverrel és szoftverrel kapcsolatban.

Az iskolai oktatásban még nem állnak kellő számban rendelkezésre szoftverek, és a meglévők sem az egységesen végiggondolt tantervek részei, hanem a meglévő tantervek egy-egy fejezetéhez írt programok. A középiskolai tanárok közül napjainkra közel 4000 fő részesült számítástechnikai alap- és továbbképzésben. Egyre nő a programokat író tanárok száma is.

Magyarországon 1983 szeptemberében indult el az iskolaszámítógép-program, amelynek eredményei, örömei és gondjai napjainkra már érzékelhetők.

A könyv azt a gondolatmenetet mutatja be, mely szerint a számítógépek oktatásbeli alkalmazása nem tantárgy, hanem elsősorban szemlélet, számítógépes műveltség. Kezdetben a tanárok egy része idegenkedett a számítógépes oktatástól, mert attól tartottak, hogy a gépek feleslegessé teszik munkájukat. Mára már világossá vált, hogy a számítógépek nem pótolják vagy helyettesítik a pedagógusok tevékenységét, hanem hatékonyan egészítik ki azt.

Pajor Gábor:
Az IBM PC-ről kezdő felhasználóknak
1. A hardver
(Budapest, 1987.
LSI ATSZ,
63 oldal. Ára: 66,— Ft.)

Ma a világon a személyi számítógépekkel dolgozók zöme felhasználó, aki munkája hatékonyságának növelésére használ számítógépet.

Ez a könyv az alkalmazókhoz szól. Célja, hogy a gépek használóit átsegítse a kezdeti lépések nehézségein. Olyan kérdésekre igyekszik választ adni, amelyek a legegyszerűbbek közé tartoznak, ezért sokszor nem is kerülnek szóba, és a géppel végül is egyedül maradt kezelőnek kell a — néha igen drága — tanulópénzt megfizetnie, vagy egyszerűen csak az idejét áldozza feleslegesen.

A könyv témái között olyanok szerepelnek, amelyek segíteni szeretnék azokat, akik még nem döntöttek, hogy milyen rendszert vásároljanak, mire legyenek tekintettel saját leendő hálózatuk kialakításában. A könyv szerzője azért választotta témájául az IBM PC XT és AT típusokat, mert ez az a két típus, amely a jelenlegi körülmények között beszerezhető és azt a teljesítményt nyújtja, amely ma a professzionális személyi számítógépektől elvárható.

A kötet a számítógép mechanikai és elektronikai részeivel foglalkozik, vagyis mindazzal, ami a gepben fizikailag jelen van. Részletesen ismerteti az IBM PC-k alapkonfigurációját, a háttértárolókat, a monitorokat, a billentyűzetet, a nyomtatókat és plottereket, a számítógép és perifériáinak összekapcsolását.

Faludi András:
A dBASE és a PROP-BASE adatbázis-kezelő rendszerek kézikönyve
(Budapest, 1987.
Számítástechnika-Alkalmazási Vállalat,
322 oldal. Ára: 197,— Ft.)

A dBASE II és a PROP-BASE adatbázis-kezelő rendszer mikroszámítógépekre készült. Ezek az adatbázis-kezelők lehetővé teszik az adatok kényelmes kezelését — adatbázisok létrehozása, azok egyszerű módosíthatósága, nagyfokú programozhatóságuk és fejlett belső automatikájuk révén.

Hazánkban a dBASE II 2.3. változata és a PROP-BASE rendszer a legelterjedtebb, így ezek teljes ismertetését tartalmazza a könyv. Mindkét adatbázis-kezelő használata középiskolai végzettség esetén minimális számítástechnikai ismeretekkel is elsajátítható. Természetesen ismerni kell hozzá az adott számítógép kezelését és tulajdonságait. A kezdők aránylag gyorsan megtanulhatnak értékes és használható programokat készíteni a könyvben közölt ismeretek alapján. Azok, akik már programoztak más nyelveken, könnyedén elsajátíthatják az adatbázis-kezelő rendszer használatát.

A könyv három fő részből áll. Az első rész ismerteti a rendszerek szerkezetét és nyelvi jellemzőit. A második rész az adatbázisok létrehozásától a feldolgozott adatok megjelenítéséig a rendszer alkalmazásának igényei szerint mutatja be a parancsok adta lehetőségeket egy-egy adott felhasználási környezetben. A harmadik rész a két adatbázis-kezelő rendszer parancsait tartalmazza alfabetikus sorrendben, részletes leírásukkal. Az első két részben megismert parancsok összes formáját megtalálhatja itt az olvasó. A függelék tartalmazza a parancsok, műveletek, beépített függvények pár szavas magyarázatát adó listát és a hibaüzenetek felsorolását.

Bátonyterenye

Bátonyterenyén, a Bartók Béla Általános Iskolában társadalmi összefogással számítástechnikai előadót alakítottak ki, ahol egyszerre 70 tanuló ismerkedhet a számítástechnika alapjaival. A terem alkalmas arra is, hogy a helyi és a körzetben tanító pedagógusok képzését, továbbképzését szolgálja.

Űrutazással Japánba

Az új űrhajó- és repülőgéptípusok megépítése ma már csak a számítástechnika fejlődésének függvénye. Ezt a felismerést igyekszik kamatoztatni az amerikai repülési és űrhajózási hivatal, mikor egyre nagyobb teljesítményű számítógépeket használ. Már a jelenlegi kapacitások egy részét is a következő évszázad repülőgépének kifejlesztésére fordítják. A várhatóan 10 éven belül elkészülő, Orient Express fantázianevű gép hiperszonikus lesz, azaz huszonötszörös hangsebességgel repül. Az Egyesült Államok és Japán között 2 óra alatt teszi meg az utat úgy, hogy közben az űrbe emelkedik.

Kereslet—kínálat

A Konjunktúra- és Piackutató Intézet a Datorg Rt.-vel közösen, a Külkereskedelmi Minisztérium támogatásával számítógépes információs rendszert hozott létre, elsősorban az exportképes termékek és a szabad termelési kapacitások felkutatására, értékesítésük elősegítésére. A június óta működő rendszerbe bevitték mindazokat az üzleti információkat, amelyekhez a külkereskedelmi kirendeltségek és különböző kiadványok útján hozzájutottak. A rendszerhez csatlakozó hazai termelővállalatok nemcsak az exportlehetőségekről tájékozódhatnak szélesebb körben, hanem felkínálhatják hasznosításra szabad kapacitásaikat is. A szervezők célja, hogy megkönnyítsék a keresletet és a kínálat egymásra találását.

NEKTÁR

A Népzenét Elemző Kottás Adattár — a NEKTÁR — létrehozásához négy évvel ezelőtt kezdték meg a program kidolgozását az Állami Gorkij Könyvtár zeneműtárában. Céljuk, hogy tízezer magyar népdalt számítógépre vigyenek. A programot könyvtárak, valamint éneket és zenét oktató intézmények használhatják majd, hogy eredményesebbé tegyék a fiatalok zenei nevelését. A NEKTÁR lehetővé teszi, hogy a népdalokat mágneslemezen tárolják, majd a kottát képernyőre kiírva elemezzék és a dallamot kívánság szerint ismételhessék. A dalokat különböző jellemzők (a forrás, a kezdő sor, a gyűjtő neve stb.) alapján lehet megkeresni. Igaz, hogy a C64 „hangja" nem olyan szép, mint a hangszereké, de a hangmagasság és a ritmus minden esetben pontos.

Tavaszra elkészült a program első része: annak a száz dalnak a feldolgozása, amely a Zeneművészeti Főiskola felvételi vizsgáihoz szükséges. A programot a Gorkij Könyvtár zeneműtárában ki lehet próbálni.

Korszerűsödik a tudakozó

A Magyar Posta mikrogépek alkalmazásával fokozatosan korszerűsíti a telefontudakozót. Nemrég készült el az első lépcső, amely a számváltozásokról adott tájékoztatást segíti és gyorsítja. A telefonon érkező érdeklődésre a gépkezelő begépeli a keresett számot, és a képernyőn megjelenő választ pillanatokon belül közölheti az ügyféllel. A Posta Számítástechnikai Intézet programozói már a következő fejlesztési lépcsőn dolgoznak: hamarosan megvalósul a név, a cím és a szám szerinti tudakozódás gépesítése is.

Négy mérnök egy üzletben

A számítástechnika felhasználói ma még kifejezetten igénylik a szakszerű tájékoztatást, ami egyrészt a szakma újszerűségének, másrészt a szinte naponta jelentkező műszaki és piaci újdonságoknak tudható be. Gondolt erre a Bizományi Áruház Vállalat is: Bécsi utcai újonnan megnyitott számítástechnikai és videoszaküzletében a vásárlókat négy, számítógép-gyártási és -alkalmazási tapasztalattal rendelkező mérnök várja, akik a tanácsadáson túl telepítésre, betanításra, az új gépeknek a már meglévő gépparkba való beillesztésére is vállalkoznak.

Az üzlet másik újszerűsége a számítógépcsere. Az elmúlt 2-3 évben sok vállalat vásárolt professzionális célokra házi számítógépet, például Commodore 64-et vagy ZX—Spectrumot. Azóta többségük „kinőtte" e gépek teljesítményét, és szeretné azokat nagyobb kapacitású, gyorsabb, korszerűbb professzionális gépekre kicserélni. Egyes iskolák, családok, magánszemélyek viszont jutányos áron szívesen vásárolnák meg ezeket a gépeket. Az üzlet napi áron átveszi a közületek „levetett" számítástechnikai eszközeit, és helyettük korszerűbb gépeket és perifériákat kínál.

Gyémántgép

A gyémánt különleges tulajdonságai miatt — a legkeményebb anyag, vegyileg szinte megtámadhatatlan, egyszerre kiváló hővezető és villamos szigetelő, kristályszerkezete olyan, mint a szilíciumkristályé — felvetődött a gyémánttranzisztor előállításának gondolata. Nagy előnye lenne hőállósága és a sugárzások iránti érzéketlensége is, különösen az űrbeli alkalmazásoknál. A kísérleteket elkezdték a Szovjetunióban, Japánban és az USA-ban. A japán Sumitomo Electric Industries már jelentette, hogy sikerült mind n-típusú, mind p-típusú gyémánt félvezetőréteget kialakítani. Az eljárás során metán és hidrogén gázkeverékben nagyfrekvenciás elektromos kisülést hoznak létre, és ezzel a molekulákat atomos szénre és hidrogénre bontják. Ezután — egyelőre még nem egészen tisztázott folyamat során — a hordozóra a szénatomok nem grafitlemezként, hanem a gyémánt tetraéderes szerkezete szerint rakódnak le.

A Junior

Hazánkban is bemutatkozott a tavaly ősszel megjelent új román mikroszámítógép, a Junior. Ez a 8 bites gép Intel 8080 típusú mikroprocesszort tartalmaz, és a CP/M operációs rendszert használja. Mint a neve is mutatja, elsősorban az ifjúság számára készült; előreláthatólag főleg iskolák és művelődési házak fogják vásárolni.

A Junior mikroszámítógép

FOTOELEKTRONIK GT NOVOTRADE

Fotoelektronik — Novotrade — GT számítógép-szervizhálózat

Az ország egész területére kiterjedő és egyre bővülő szervizhálózatunk vállalja személyi és professzionális számítógépek, valamint prerifériáik garanciális és térítéses javítását, karbantartását.

Kedvező feltételekkel kötünk átalánydíjas JAVÍTÁSI ÉS KARBANTARTÁSI szerződést!

commodore típusú (C16, C64, Plus/4, C128, C610, C720, PC—10/20), valamint Atari, Sinclair Spectrum, Primo, HT és más számítógépek, Epson, Seikosha printerek és egyéb perifériák szervizelését, javítását bízza szervizmérnökeinkre, szakembereinkre!

Szervizeink:

Budapest V., Magyar u. 12—14. Telex: 22-7621	173-551
Budapest VIII., Szigony u. 9.	343-153
Pécs, Kolozsvár u. 20.	72/11-812
Miskolc, Fazekas u. 4.	46/17-011
Szeged, Székelysor 13.	62/13-377
Békéscsaba, Bartók Béla u. 37.	66/27-195
Szombathely, Szalonok u. 31.	94/14-519
Debrecen, Holló L. u. 14.	

Tekintse meg az ELEKTROSOFT Kisszövetkezet bemutatóját a SW '88-on!

ÉPÍTŐIPAR
- költségvetés-készítés
- elemzés
- tervezés

HÚSIPAR
- ügyviteli csomag
- élőállat-gazdálkodás

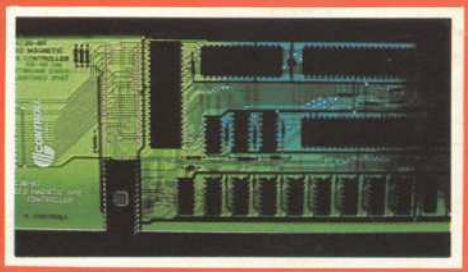
ÁLTALÁNOS VÁLLALATI INFORMÁCIÓS RENDSZEREK
- főkönyv (kartonkezeléssel)
- folyószámla (kartonkezeléssel)
- anyagkönyvelés
- készletgazdálkodás

ALAPSZOFTVER
- szubrutincsomag C-hez
- konkurens IS fájlkezelő NOVELL és MP/M alá
- C—ISAM interfésszel

A SLÁGER: 8 MUNKAHELYES, CSOPORTOS ADATRÖGZÍTŐ

Valamennyi termékünk portábilis. A SW '88-on megtekintheti, hogyan működik ugyanaz a 8 bites technikán és NOVELL hálózaton. Referenciahelyeink képviselői is az Önök rendelkezésére állnak.

ÖSSZEÁLL A KÉP...



A SOFTWARE '88 kiállításon

november 10. és 13. között

Újdonságok és jól bevált rendszerek a hálózatos formában működő szoftverek között.

Bemutatjuk a CONTROLL hardverein a Duna Inter'-Continental Szállodában.

Ha talál megfelelő programot, gépet is azonnal tudunk szállítani kiegészítő egységekkel és különleges perifériaillesztőkkel.

ÍGY VALÓBAN ÖSSZEÁLL A KÉP!

CONTROLL
ELEKTRONIKAI ÉS SZÁMÍTÁSTECHNIKAI KISSZÖVETKEZET

Budapest, Szász K. u. 2. 1027
Tel.: 158-430; 158-428